Learning the Coefficients: A Presentable Version of Border Complexity and Applications to Circuit Factoring

C. S. Bhargav
IIT Kanpur
Kanpur, India
bhargav@cse.iitk.ac.in

Prateek Dwivedi
IIT Kanpur
Kanpur, India
pdwivedi@cse.iitk.ac.in

Nitin Saxena
IIT Kanpur
Kanpur, India
nitin@cse.iitk.ac.in

ABSTRACT

The border, or the approximative, model of algebraic computation ($\overline{\mathsf{VP}}$) is quite popular due to the Geometric Complexity Theory (GCT) approach to P ≠ NP conjecture, and its complex analytic origins. On the flip side, the definition of the border is inherently *existential* in the field constants that the model employs. In particular, a poly-size border circuit $C(\varepsilon, \boldsymbol{x})$ cannot be compactly presented in reality, as the limit parameter ε may require *exponential* precision. In this work we resolve this issue by giving a constructive, or a *presentable*, version of border circuits and state its applications.

We make border presentable by restricting the circuit C to use only those constants, in the function field $\mathbb{F}_q(\varepsilon)$, that it can generate by the ring operations on $\{\varepsilon\} \cup \mathbb{F}_q$, and their division, within poly-size circuit. This model is more expressive than VP as it affords exponential-degree in ε; and analogous to the usual border, we define new border classes called $\overline{\mathsf{VP}}_\varepsilon$ and $\overline{\mathsf{VNP}}_\varepsilon$. We prove that both these (now called *presentable* border) classes lie in VNP. Such a 'debordering' result is not known for the classical border classes $\overline{\mathsf{VP}}$ and respectively for $\overline{\mathsf{VNP}}$. We pose $\overline{\mathsf{VP}}_\varepsilon = \overline{\mathsf{VP}}$ as a new conjecture to study the border.

The heart of our technique is a newly formulated *exponential interpolation* over a finite field, to bound the Boolean complexity of the coefficients before deducing the algebraic complexity. It attacks two factorization problems which were open before. We make progress on (Conj.8.3 in Bürgisser 2000, FOCS 2001) and solve (Conj.2.1 in Bürgisser 2000; Chou,Kumar,Solomon CCC 2018) over all finite fields:

(1) Each poly-degree irreducible factor, with multiplicity coprime to field characteristic, of a poly-size circuit (of possibly *exponential*-degree), is in VNP.
(2) For *all* finite fields, and *all* factors, VNP is closed under factoring. Consequently, factors of VP are *always* in VNP. The prime characteristic cases were open before due to the inseparability obstruction (i.e. when the multiplicity is not coprime to q).

CCS CONCEPTS

• **Theory of computation** → **Algebraic complexity theory**; **Complexity classes**; **Circuit complexity**.

KEYWORDS

approximative, border, presentable, deborder, factoring, closure, circuits, GCT, VP, VNP

ACM Reference Format:
C. S. Bhargav, Prateek Dwivedi, and Nitin Saxena. 2024. Learning the Coefficients: A Presentable Version of Border Complexity and Applications to Circuit Factoring. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24), June 24–28, 2024, Vancouver, BC, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3618260.3649743

1 INTRODUCTION

The notion of "approximation" is a powerful idea in theoretical computer science, both in designing algorithms for problems and in analyzing their computational hardness. In Valiant's framework of algebraic complexity theory [78, 79], the *border complexity* of a polynomial measures how efficiently it can be approximated. In this framework, a multivariate polynomial is computed by a non-uniform model called an *algebraic circuit* – a directed acyclic graph with internal nodes labeled by + and × operators, leaves labeled by variables or constants from the underlying field $\mathbb{F}$, and a designated output node. The circuit computes an n-variate polynomial $f(\boldsymbol{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ in a natural bottom-up way. Computing a polynomial by a circuit always refers to computing a family of polynomials $\{f_n\}$, one for each $n \in \mathbb{N}$.

The measure of efficiency is the number of vertices and edges of the graph, which we refer as the size of the circuit. We denote the size of the smallest circuit over $\mathbb{F}$ computing the polynomial f by $\mathrm{size}_{\mathbb{F}}(f)$. Valiant [78] hypothesized that there are *explicit* polynomials that cannot be computed by circuits of small size. It is formalized as what we now call the VP ≠ VNP conjecture. The class VP (Valiant's P) consists of all polynomials with degree bounded by a polynomial in the number of variables n (=: poly(n)), which can be computed by algebraic circuits of size poly(n). An algebraic analogue of NP was defined as well using an exponential sum of VP polynomials. More formally,

Definition 1.1 (Valiant's NP*).* The class VNP is the set of all polynomials $f \in \mathbb{F}[x_1, \ldots, x_n]$ such that there exists a polynomial $g \in \mathbb{F}[x_1, \ldots, x_n, y_1, \ldots, y_m]$ in VP with $m = \mathrm{poly}(n)$ and

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{a} \in \{0,1\}^m} g(\boldsymbol{x}, \boldsymbol{a}).$$

We call $y_1, \ldots, y_m$ the *witness* (or *hypercube*) variables and $g(\boldsymbol{x}, \boldsymbol{y})$ as the *verifier circuit.* It is straightforward to see that VP ⊆ VNP, and

STOC '24, June 24–28, 2024, Vancouver, BC, Canada

ACM ISBN 979-8-4007-0383-6/24/06
https://doi.org/10.1145/3618260.3649743

Valiant's conjecture is that the inclusion is strict. The surveys of [17, 52, 66, 73] provide an excellent overview of algebraic complexity and the current state of lower bounds. For a more extensive but slightly dated treatment, see [12, 13].

1.1 Algebraic Approximation

There is a natural way to associate a Euclidean (or Zariski) topology with the polynomial ring. This confers a notion of limit and, thereby, a way of approximating a polynomial by a sequence of polynomials (see, e.g., [10, Section 2.3]). The topological notion has been extensively studied in the context of designing algorithms for matrix multiplication [6, 7, 22, 50, 75]. However, in Valiant's framework, the simplest definition for *algebraic* approximation and border complexity was given by Bürgisser [14]. We say that a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ is approximated by a polynomial $g \in \mathbb{F}[\varepsilon][x_1, \ldots, x_n]$ to an *order of approximation* M if $g(\boldsymbol{x}, \varepsilon) = \varepsilon^M f(\boldsymbol{x}) + \varepsilon^{M+1} Q(\boldsymbol{x}, \varepsilon)$, for some $Q \in \mathbb{F}[\varepsilon][x_1, \ldots, x_n]$. The *border size* of f denoted by $\overline{\text{size}}(f)$, is defined as $\text{size}_{\mathbb{F}[\varepsilon]}(g)$,the size of the polynomial g over the ring $\mathbb{F}[\varepsilon]$ (instead of being over the *constants* $\mathbb{F}$).

Note that $\lim_{\varepsilon \to 0} \varepsilon^{-M} g(\boldsymbol{x}, \varepsilon) = f(\boldsymbol{x})$. Furthermore, arbitrary polynomials in ε are treated as 'free constants' in the circuit of g. Alternately, we can also consider the approximating polynomial g over the rational function field $\mathbb{F}(\varepsilon)$ (as done in our paper abstract) and aim for an approximation of the form $g' = f + \varepsilon Q$, with the effect of $\lim_{\varepsilon \to 0} g' = f$. It is not hard to see via scaling arguments that these notions are equivalent, in particular $g' := \varepsilon^{-M} g$. For a discussion of the different notions of approximation and their equivalence, see [14, Lemma 5.6], [11, Section 2], and also [60, Theorem 2.33].

As a natural extension, we can define the *approximate closure* of VP, called $\overline{\text{VP}}$ as the set of poly(n)-degree polynomials whose *border size* is bounded by poly(n). Clearly, $\text{VP} \subseteq \overline{\text{VP}}$. In an ambitious program to resolve the $\text{P} \overset{?}{=} \text{NP}$ question using methods from algebraic geometry and representation theory, Mulmuley and Sohoni [58] strengthened Valiant's conjecture by postulating that VNP is not contained in $\overline{\text{VP}}$ [1]. Their proposal (detailed further in [59]) was to use techniques from representation theory to prove lower bounds on border complexity. For expository references on the GCT program, see [10, 49, 56, 57, 65].

Completely independently and almost at the same time, Bürgisser [14, 15] introduced and used border complexity to factor multivariate polynomials. Factorization is a very basic notion in algebra, and a complexity class is 'well behaved' in some sense if it is closed under factorization. In a string of highly influential papers [38–40, 43], Kaltofen showed that over fields of characteristic zero, the class VP is closed under taking factors (also see [42]). In fact, if a polynomial factorizes as $f = u^e v$ with u and v co-prime, then Kaltofen [39] showed that u can be computed by a circuit of size poly$(e, \deg(u), \text{size}(f))$. One might expect, for exponential-degree f, that the size of u depends only on its degree and the size of f, and that the dependence on multiplicity e can be completely removed. In other words, we expect that any poly(n)-degree factor of a poly(n)-size circuit (with no restrictions on degree) is in VP. This is known as the Factor Conjecture [13, Conjecture 8.3]. In his work, Bürgisser [14] showed that for border complexity, the factor conjecture is indeed true – the factor u above, is in $\overline{\text{VP}}$. This makes factor conjecture an important stepping-stone towards understanding algebraic computation. Our work will build on this theme.

1.2 Our Goal: To Make Border Presentable

The notion of approximation in Valiant's framework arose at the same time in different contexts. This suggests that it is indeed very natural. But a basic question, made even more pertinent by the discussion above, that remains open to this day is whether approximation bestows more computational power, or in other words, whether $\text{VP} \overset{?}{=} \overline{\text{VP}}$ [14, Problem 4.3]. In a recent work on border complexity [23] a more general question was asked, called *de-bordering*. Given a polynomial $f \in \overline{C}$ in the approximate closure of a class C, what is an upper bound on the exact (non-approximate) complexity of f? A class C is *border-closed* if $C = \overline{C}$ Although one might expect a class to not differ too much from its border class, it is far from clear since, in the definition of approximation, we allow arbitrary polynomials in ε of arbitrary complexity to be used as free constants. This arbitrariness makes the definition of approximation inherently *existential*. In fact, we do not even know whether $\overline{\text{VP}}$ is contained in VNP.

As a way of making approximation more constructive, while retaining its essence, in this work we propose and study a natural restriction on the definition of approximation, that we call *presentability*. The *presentable* class $\overline{\text{VP}}_\varepsilon$ is the same as $\overline{\text{VP}}$ but with the additional condition that all the polynomials in ε used as 'constants' in the approximating circuit $g(\boldsymbol{x}, \varepsilon)$, have polynomial-size circuits themselves (see Definition 4.3).

There has previously been an attempt via 'degenerations' [33] to identify a subclass of $\overline{\text{VP}}$ that is explicit. In what they term *p-definable one-parameter degeneration*, the authors restrict the coefficients of the ε-polynomials to be generated using circuits in VP. Our presentable border is a more natural version of $\overline{\text{VP}}$ and *cannot* be obtained as a p-definable degeneration of VP, making our notion incomparable to the concept of degeneration as studied in [33]. We can extend our concept of presentable border to $\overline{\text{VNP}}_\varepsilon$ over any field $\mathbb{F}$.

Definition 1.2 (Presentable $\overline{\text{VNP}}$). The presentable border class $\overline{\text{VNP}}_\varepsilon$, over $\mathbb{F}$, is defined as the set of polynomials $f \in \mathbb{F}[x_1, \ldots, x_n]$ such that there is an approximating polynomial $g \in \mathbb{F}[\varepsilon][x_1, \ldots, x_n]$ expressing

$$g(\boldsymbol{x}, \varepsilon) \ =: \ \varepsilon^M f(\boldsymbol{x}) + \varepsilon^{M+1} Q(\boldsymbol{x}, \varepsilon) \ ,$$

for some *error* $Q \in \mathbb{F}[\varepsilon][x_1, \ldots, x_n]$ and *order* $M \in \mathbb{N}$; moreover, there exists a *verifier* polynomial $h \in \mathbb{F}[x_1, \ldots, x_n, y_1, \ldots, y_m, \varepsilon]$, with $m, \deg_{\boldsymbol{x},\boldsymbol{y}}(h)$ and $\text{size}_{\mathbb{F}}(h)$ all bounded by poly(n), satisfying a *hypercube-sum* expression

$$\sum_{\boldsymbol{a} \in \{0,1\}^m} h(\boldsymbol{x}, \boldsymbol{a}, \varepsilon) \ = \ g(\boldsymbol{x}, \varepsilon) \ .$$

The pair $(m, \text{size}_{\mathbb{F}}(h))$ constitutes the size parameters for the polynomial family $f = f_n$ in $\overline{\text{VNP}}_\varepsilon$. Crucially, although the bound

[1] More precisely, they conjectured that the padded Permanent does not lie in the orbit closure of small Determinants.

on $\mathrm{size}_{\mathbb{F}}(h)$ (instead of $\mathrm{size}_{\mathbb{F}[\varepsilon]}(h)$) constrains the ε-polynomials to have small circuits, we do not restrict the degree of ε, which could be exponential in $\mathrm{size}_{\mathbb{F}}(h)$. This makes this new class potentially harder than VNP. It is easy to see that $\mathsf{VNP} \subseteq \overline{\mathsf{VNP}}_{\varepsilon} \subseteq \overline{\mathsf{VNP}}$. But, it is not clear whether these containments are strict. Similarly, the containment $\mathsf{VP} \subseteq \overline{\mathsf{VP}}_{\varepsilon} \subseteq \overline{\mathsf{VP}}$ raises new questions.

1.3 Our Results

Our first main result is the *de-bordering* of the presentable border classes.

Theorem 1.3 (Presentable is Explicit). *Over* any *finite field,* $\overline{\mathsf{VNP}}_{\varepsilon} = \mathsf{VNP}$.

This gives us an interesting tower of containments $\mathsf{VP} \subseteq \overline{\mathsf{VP}}_{\varepsilon} \subseteq \mathsf{VNP}$. In addition, it yields a generalization of Valiant's conjecture to all presentable models.

Conjecture 1.4. $\mathsf{VP} = \overline{\mathsf{VP}}_{\varepsilon} \neq \mathsf{VNP}$.

As a consequence of our debordering result, we make progress toward the aforementioned Factor Conjecture [13, Conjecture 8.3]. As noted earlier, Bürgisser showed that any poly(n)-degree factor of a poly(n)-size circuit is in $\overline{\mathsf{VP}}$. We observe that it is in fact in $\overline{\mathsf{VP}}_{\varepsilon}$, and thus by Theorem 1.3 in VNP.

Corollary 1.5 (Debordering factors). *Let f_n be a n-variate polynomial family over a finite field that has a* poly(n)*-degree irreducible factor u_n of multiplicity co-prime to the characteristic of the field. If* size(f_n) *is* poly(n)*, then u_n is in* VNP.

Remark. A few points of note:

(1) The $\deg(f_n)$ and hence, the multiplicity of u_n are possibly exponential in n. This is what makes standard factoring algorithms hopelessly inefficient.
(2) We get an *explicitness* (VNP) result for the factors, instead of a factoring algorithm. Nevertheless, it is concrete evidence supporting the factor conjecture.

Bürgisser [13, Conjecture 2.1] asked if the class VNP is closed under factorization. Over fields of characteristic zero, Chou, Kumar and Solomon [21] showed that this is indeed true. Inspired by the proof technique of Theorem 1.3, in our second main result, we use similar techniques to prove that VNP closure under factoring holds over finite fields as well, thus settling Bürgisser's conjecture in an important regime.

Theorem 1.6 (Factor closure). *Over* any *finite field, the class* VNP *is closed under factorization.*

Remark. As a corollary of the above theorem, we find that over finite fields, the factors of polynomials in VP are in VNP. This partially answers the question [13, Problem 2.1] whether VP is closed under taking factors over fields of positive characteristic. Recall that over fields of characteristic zero, we already know this to be true from the works of Kaltofen; but those methods fail in finite fields.

2 PROOF OUTLINE

We now outline the ideas and techniques used to prove our results. We will also discuss related previous work and its limitations.

2.1 Efficacy of Presentable Border

A major obstacle to de-bordering any class is that the expression for approximating a polynomial f

$$g(\boldsymbol{x}, \varepsilon) = \varepsilon^{M} f(\boldsymbol{x}) + \varepsilon^{M+1} Q(\boldsymbol{x}, \varepsilon),$$

says very little about the complexity of the ε-constants involved, which could be huge. A natural idea to isolate f from the above expression is via *interpolation* on the ε variable. This seems hard to do as apriori, the degree of ε in the polynomial g could be arbitrarily large. Already in the foundational work, Bürgisser [14, Theorem 5.7] showed that over algebraically closed fields, the order of approximation M is at most exponential in $\overline{\mathrm{size}}(f) := \mathrm{size}_{\mathbb{F}[\varepsilon]}(g)$, the *border size* of the polynomial f. Therefore, moving to presentable border classes $\overline{\mathsf{VP}}_{\varepsilon}$ and $\overline{\mathsf{VNP}}_{\varepsilon}$ does not lead to any ε-degree loss, since they allow for an exponential degree in ε. But unless one can show a *polynomial* bound on the order of approximation [2], interpolation seems to give a bound of the form $\mathrm{size}(f) \leq \exp(\overline{\mathrm{size}}(f))$.

Known debordering results. Incidentally, the known debordering results for restricted models of computation seldom use interpolation. In the workshop seminar [27], Forbes remarked that Nisan's characterization implies the closure of ROABPs or equivalently non-commutative ABPs (see [29, Chapter 4] for definitions and [8, Lemma 5.2] for the proof). Using structural properties of computational models and monotonicity, it can be shown that almost all the interesting *monotone* complexity classes are border-closed [9, 16]. We also know of certain cases where a class is strictly contained in its closure. Elementary but clever matrix identities reveal that closure of width-2 algebraic branching programs is the same as the closure of general formulas [11]. Together with the results of [2, 3], this implies that width-2 algebraic branching programs are *not* border-closed!

In a similar vein, Kumar [46] showed that the closure of bounded top-fanin (exponential size) depth-3 circuits is *universal* whereas there are polynomials that cannot be computed by their 'classical' counterparts, regardless of the size [18, 46]. A recent work of Dutta, Dwivedi and Saxena [23] introduced the DiDIL technique and showed that every polynomial in the closure of bounded top-fanin depth-3 circuits has a polynomial sized algebraic branching program. Building on that, Dutta and Saxena [24] showed an *exponential* separation between consecutive border classes $\overline{\sum^{k}\prod\sum}$ and $\overline{\sum^{k+1}\prod\sum}$. Unfortunately, these de-bordering and separation results are based on characterizations and properties of restricted classes that are not known for general classes such as $\overline{\mathsf{VP}}_{\varepsilon}$ and $\overline{\mathsf{VNP}}_{\varepsilon}$.

Adapting interpolation to presentable border. Surprisingly, although interpolation seemed unhelpful on first glance, we show that a structural modification does indeed help in de-bordering when we move to presentable border classes. Note that $\mathsf{VNP} \subseteq \overline{\mathsf{VNP}}_{\varepsilon}$ by definition. For the other direction, to show the containment in VNP, instead of directly using the definition, we turn to the following criterion of Valiant [78] (also see [13, Prop. 2.20]) which

[2] See [11, Corollary 3.10] for an example of debordering through interpolation when a related measure of approximation called 'error degree' is *polynomially* bounded.

essentially states that *low-degree* polynomials whose coefficients are *effectively computable* in the boolean world are in VNP in the algebraic world. Here, we state a version that works over all fields. For a mathematical object a, we denote its *boolean* encoding by $\langle a \rangle$.

Proposition 2.1 (Valiant's criterion). *Let f be a polynomial in n variables of degree* $\mathrm{poly}(n)$ *over a field $\mathbb{F}$ such that $f = \sum_{\boldsymbol{e}} c_{\boldsymbol{e}} \boldsymbol{x}^{\boldsymbol{e}}$. Suppose that there exists a string function $\phi : \{0,1\}^* \mapsto \{0,1\}^*$ in* #P/poly *such that $\phi(\langle \boldsymbol{e} \rangle) = \langle c_{\boldsymbol{e}} \rangle$. Then, the polynomial f is in* VNP *over the field $\mathbb{F}$.*

Remark. Unlike the usual definition of #P which consists of functions mapping $\{0,1\}^*$ to $\mathbb{N}$, we find it more convenient to consider functions that output binary strings. Coefficients are usually elements of a finite field $\mathbb{F}_q$ of size p^a (say). Each element in $\mathbb{F}_q$ is a univariate polynomial of degree less than a with coefficients from $\mathbb{F}_p$ (see [71, Chapter 19] and [55]). Since $\mathbb{F}_p$ is isomorphic to $\mathbb{Z}$ mod p, we treat each element of $\mathbb{F}_q$ as a list of a integers encoded as a string of length $O(a \log p)$.

Over finite fields we use a weaker version of the criterion in our proofs, where instead of assuming coefficient function $\phi \in$ #P/poly, we assume $\phi \in \#_p$P/poly [13, Section 4.3]. Formally that means, there exists a function $\psi \in$ #P/poly such that $\phi(\langle \boldsymbol{e} \rangle) = \psi(\langle \boldsymbol{e} \rangle) \bmod p$ [3]. We omit this subtlety wherever it is clear from the context. Refer Section 3 and the full version of the paper [4] for various helpful definitions of counting classes.

Consider now a polynomial $f = \sum_{\boldsymbol{e}} c_{\boldsymbol{e}} \boldsymbol{x}^{\boldsymbol{e}}$ in $\overline{\mathsf{VNP}}_\varepsilon$ over the finite field $\mathbb{F}_q$. We would like to show that the coefficient function $\phi : \langle \boldsymbol{e} \rangle \mapsto \langle c_{\boldsymbol{e}} \rangle$ is in #P/poly. We have access to f only using the approximating polynomial g

$$g(\boldsymbol{x}, \varepsilon) = \varepsilon^M f(\boldsymbol{x}) + \varepsilon^{M+1} Q(\boldsymbol{x}, \varepsilon),$$

which is of the following hypercube-sum form

$$g(\boldsymbol{x}, \varepsilon) = \sum_{\boldsymbol{a} \in \{0,1\}^m} h(\boldsymbol{x}, \boldsymbol{a}, \varepsilon),$$

for some verifier circuit $h \in \mathbb{F}_q[x_1, \ldots, x_n, y_1, \ldots, y_m, \varepsilon]$, whose degree in the variables $\boldsymbol{x}$ and $\boldsymbol{y}$ is bounded by $\mathrm{poly}(n)$. Note that h is *not* in VP since its degree in ε can be exponential in n.

We will extract the coefficient of $\varepsilon^M \boldsymbol{x}^{\boldsymbol{e}}$ in g by carefully choosing the interpolation points to be roots of unity, whose (multiplicative) *order* is 'only' exponential. Consequently, we show that the coefficient $c_{\boldsymbol{e}}$ can be obtained as a hypercube sum of an *exponential degree* algebraic circuit of *polynomial size* (Lemma 4.1) We enumerate two tricky issues that are handled in the proof.

(1) It would not be possible to control the size of this extraction circuit (over the underlying field $\mathbb{F}_q$) if we were to use the usual definition of $\overline{\mathsf{VNP}}$, mainly because the ε-constants might truly require exponential *size* circuits. Working with $\overline{\mathsf{VNP}}_\varepsilon$ lets us keep the circuit size small while retaining the exponentially large degree of ε.
(2) The choice of interpolation points must be careful; otherwise, just to write down the interpolation formula, we would need to invert an exponentially large matrix of *generic* constants, which would again require circuits of exponential size. In addition, we need the various points to eventually map to a suitable hypercube $\{0,1\}^\ell$, which places further constraints on the design of the points.
We solve these problems by using the properties of finite fields that allow us to transfer to a much better-behaved Boolean computation model. In particular, we use a multiplicative generator ω of an exponentially large field $\mathbb{F}_{q'}$ to realize the hypercube points.

Using finite field arithmetic and the closure of the Boolean class #P under exponential sums, we move from the algebraic world to the Boolean one (Lemma 4.2). Thus, we show that the algebraic circuit above (from Lemma 4.1) can be simulated by a (multi-output) Boolean circuit of polynomial size; furthermore, the hypercube sum computing the coefficient function is demonstrated in #P/poly. Valiant's criterion (Proposition 2.1) now implies that the polynomial f is indeed in VNP.

2.2 Factor Closure over Finite Fields

The two classical paradigms involved in factoring multivariate polynomials are *Hensel lifting* and *Newton iteration* (see, e.g. [80, 81]), which have historical origins in complex analysis. Since the foundational results of Kaltofen on *uniform* closure of the class VP under taking factors, variants of these techniques have been used successfully to study factors of classes inside VP, such as sparse polynomials [5, 32, 51, 82], polynomials with bounded-depth circuits [26, 61] and bounded individual degree [61], algebraic branching programs [36, 41, 74] and even classes beyond VP such as VNP [21] and polynomials of exponential degree [25], not only to show closure results, but also to provide factoring algorithms. There have been many proofs of the original VP closure result itself. See [13, 20, 25, 45, 61] for some alternate ones.

Various proofs and techniques introduced in these works have evolved to provide applications in various areas of computer science, for instance hardness-randomness tradeoffs [1, 21, 26, 34, 37, 47, 48], polynomial identity testing [45, 72], coding theory [35, 76], cryptography [19], proof complexity [31], convex optimization [62] and more. See [28, 69] for an introduction and survey of polynomial factoring.

In a recent work, VNP was proved to be closed under factoring over fields of characteristic zero [21]. A crucial step in their proof, which involves approximating a root of a polynomial to increasingly higher precision using Newton iteration, fails to work over finite fields (a more important case in computer science applications). To prove that the class VNP is closed under factoring over fields of positive characteristic p, we reduce the problem to two cases. Let f be a polynomial in VNP. Following [20], we have one of the following:

(1) The polynomial $f = u^e$ is a power of a factor u.
(2) The polynomial $f = u \cdot v$ is a product of co-prime polynomials u and v.

We would like to show that the factor u is in VNP in both cases. The proof of Case 2 (Lemma 5.3) uses slight modifications of standard techniques developed over the years [21, 39, 45]. We first transform the polynomial so that it is monic and bi-variate. We start the Hensel lifting process with two coprime univariate factors and lift them to high enough precision (with respect to a degree measure). We use a version of the lift that automatically gives us

[3] In a slight abuse of notation, we assume the function ψ maps $\{0,1\}^*$ to $\mathbb{N}$.

the factors at the end. To finally show that the factor we obtain is in VNP, we use a one-shot analysis as in [21].

Over fields of characteristic zero, it can be shown that proving Case 2 is sufficient (see proof of [20, Lemma 1.3]). However, in a finite field $\mathbb{F}_q$, this reduction only works if the characteristic p of the field does not divide the exponent e (we can call this the *separable* case). Our main contribution is showing that if $f = u^{p^k}$ for some $k \geq 1$, then u is in VNP (Lemma 5.2). Using this result, we can then handle all powers (Lemma 5.4).

All previous known techniques fail in the case where the exponent e is a prime power. Inspired by the proof of Theorem 1.3, we take a completely different approach. Consider the simple case where $f = u^p$. The coefficients of u and coefficients of f are related by a simple Frobenius action. It turns out that Valiant's criterion (Proposition 2.1) for a polynomial being in VNP also has a converse (Lemma 5.5). It was remarked in [54, Section 6] that the fact has been observed before in [64], though we could not find a written reference [4]. We give an independent proof for finite fields in this paper by first noting that any coefficient of a VNP polynomial can be obtained as a hypercube-sum of evaluations of a VP circuit. Next, we use ideas similar to the proof of Theorem 1.3 to convert the algebraic expression thus obtained to a Boolean #P/poly circuit.

Since $f \in$ VNP, the inverse of Valiant's criterion gives us that its coefficient function is in #P/poly. We obtain the coefficients of u by performing an *inverse* Frobenius transformation, which we demonstrate in #P/poly. Finally, using Valiant's criterion in the forward direction, we see that the factor u is in VNP.

3 PRELIMINARIES

Throughout the paper, we refer and use well-known structural results of Algebraic Complexity Theory. In this section we will formally state them, and provide relevant references to comprehensive discussion on them.

Homogenisation. For a degree-d polynomial f, we denote its degree-k homogeneous components by $\text{Hom}_k(f)$. Similarly, we define $\text{Hom}_{\leq k}(f)$ equal to $\sum_{i\in[k]} \text{Hom}_k(f)$. The following well-known structural result proves that given a blackbox access to a circuit computing the polynomial f, we can construct a circuit that computes all its homogeneous components. Refer [73, Theorem 2.2] for the proof.

Lemma 3.1 (Homogenisation). *Consider an n-variate polynomial $f := \sum_{i\in[d]} c_i(\boldsymbol{y})x^i$ computable by a circuit of size s over $\mathbb{F}$. Then $\text{size}(c_i)$ is at most $\text{poly}(s,n,d)$, for all $i \in [d]$. Moreover, $\text{size}(\text{Hom}_{\leq d}(f))$ is at most $\text{poly}(s,n,d)$.*

We will invoke the lemma to homogenise the Hensel lifting circuit, constructed in the proof of Lemma 5.9.

Hypercube-sum of Formulas. An algebraic circuit is called a *formula*, if the underlying graph is a tree. In Definition 1.1 we defined the class VNP as hypercube-sum of small size *circuits*. Valiant proved in [79] that these polynomials can be equivalently computed by a hypercube-sum of small size formulas. Refer [13, Theorem 2.13] and [54, Theorem 2] for the proof. A direct consequence of the equivalence is the following structural lemma, that helps in proving closure properties of VNP in Lemma 5.1.

Lemma 3.2 (Verifier formula). *Consider an n-variate polynomial f of degree d computable by a* circuit *of size s over $\mathbb{F}$. Then, there is a verifier polynomial h, with m and the* formula *size of h both bounded by* $\text{poly}(s,n,d)$*, satisfying the hypercube-sum expression*

$$\sum_{\boldsymbol{a}\in\{0,1\}^\ell} h(x_1,\ldots,x_n,a_1,\ldots,a_\ell) = f.$$

Counting and functional complexity classes. We will review some of the computational complexity classes used in our proofs and discuss some standard closure results. For details refer to [14, Section 4.3] and [44, Section 2.2]. For a more comprehensive discussion refer to [63]. For a natural number r, $\langle r\rangle \in \{0,1\}^*$ denotes the binary encoding of r.

Definition 3.3 (#P and FP). The complexity class #P is defined as the set of string functions $\psi : \{0,1\}^* \to \{0,1\}^*$ such that there is a language $\chi \in$ P satisfying $\psi(\boldsymbol{x}) = \langle |S| \rangle$ where

$$S = \left\{\boldsymbol{y} \in \{0,1\}^{\text{poly}(|x|)} \;:\; (\boldsymbol{x},\boldsymbol{y}) \in \chi\right\}.$$

Further, a function ψ is in FP if there exists a Turing machine that computes $\psi(\boldsymbol{x})$, for all inputs $\boldsymbol{x} \in \{0,1\}^*$, in time $\text{poly}(|\boldsymbol{x}|)$.

It is easy to show that FP is contained in #P (refer [70, Lemma 8]). For finite fields $\mathbb{F}_q$ where $q = p^a$, a more useful class for our proofs is $\#_p$P.

Definition 3.4 ($\#_p$P). The complexity class $\#_p$P is defined as the set of functions $\psi : \{0,1\}^* \to \{0,1\}^*$ such that there exists a function $\Psi \in$ #P satisfying

$$\psi(\boldsymbol{x}) \equiv \Psi(\boldsymbol{x}) \mod p.$$

Any counting class can be extended to its corresponding non-uniform version where the functions accept an advice string, in addition to the input string, for computation.

Definition 3.5 (Non-uniform complexity classes). The complexity class C/poly is defined as the set of functions $\phi : \{0,1\}^* \to \{0,1\}^*$ such that there exists a ψ in class C and a polynomial length advice function $\alpha : \mathbb{N} \to \{0,1\}^*$ satisfying $\phi(\boldsymbol{x}) = \psi(\boldsymbol{x}, \alpha(|\boldsymbol{x}|))$.

We remark that the advice function α in the definition above only depends on the length of the input. Moreover, for all $n \in \mathbb{N}$, $|\alpha(n)| \leq \text{poly}(n)$.

4 PRESENTABLE IS EXPLICIT

To prove Theorem 1.3, we will begin by stating two essential lemmas of our paper which will help us in designing *effective* coefficient functions of large degree polynomials. The following lemma shows that the polynomials computable by the hypercube-sum of small sized circuits are 'closed' under coefficient extraction, i.e. there is a similar algebraic expression for each coefficient. This is like interpolation, but as the degree and number of monomials is exponential, we desire to achieve an algebraic expression that is well structured.

Lemma 4.1 (Exponential interpolation). *Let $s := \text{poly}(r, \log q)$ and $g = \sum_{\boldsymbol{e}} c_{\boldsymbol{e}}\boldsymbol{y}^{\boldsymbol{e}}$ be an r-variate polynomial over $\mathbb{F}_q$ of degree*

[4] Perifel communicated to us a proof that over $\mathbb{Q}$, the coefficients of constant-free VNP families (see [53]) are in GapP/poly.

$D := \exp(s)$ such that $g = \sum_{\boldsymbol{a} \in \{0,1\}^m} h(\boldsymbol{y}, \boldsymbol{a})$ for some polynomial h with $m, \mathrm{size}(h) \leq s$.

Then, taking $\boldsymbol{e}$ as input there exists a polynomial $t_{\boldsymbol{e}}$ over a finite field extension $\mathbb{F}_{q'}$, $q' \leq \mathrm{poly}(D)$, such that the coefficient $c_{\boldsymbol{e}} = \sum_{\boldsymbol{b} \in \{0,1\}^\ell} t_{\boldsymbol{e}}(b_1, \ldots, b_\ell)$, where ℓ and $\mathrm{size}(t_{\boldsymbol{e}})$ are at most $\mathrm{poly}(s)$.

In the subsequent lemma we show that the resulting hypercube sum above can be converted into a boolean function in #P/poly (refer the remark following Proposition 2.1). The two lemmas together build up the correct setup to invoke Valiant's criterion. Recall $s = \mathrm{poly}(r, \log q)$.

Lemma 4.2 (Algebraic to boolean complexity). *Consider an exponent vector $\boldsymbol{e} \in \{0, \ldots, D\}^r$, and let the coefficient of $\boldsymbol{y}^{\boldsymbol{e}}$ in $g \in \mathbb{F}_q[y_1, \ldots, y_r]$, denoted by $c_{\boldsymbol{e}}$, be computable by a polynomial $t_{\boldsymbol{e}}$ over a finite field extension $\mathbb{F}_{q'}$, $q' \leq \mathrm{poly}(D) \leq 2^{O(s)}$, as follows:*

$$c_{\boldsymbol{e}} = \sum_{\boldsymbol{b} \in \{0,1\}^\ell} t_{\boldsymbol{e}}(b_1, \ldots, b_\ell) \,, \tag{1}$$

where ℓ and $\mathrm{size}(t_{\boldsymbol{e}})$ are at most $\mathrm{poly}(s)$. Then, with s as the input-size parameter, there exists a function ϕ_g in #P/poly that computes $\phi_g(\langle \boldsymbol{e} \rangle) = \langle c_{\boldsymbol{e}} \rangle$.

We will defer the proof of the lemmas to the full version of our paper [4]. Meanwhile, we will use the technical lemmas to give the complete proof of our first main result.

Proof of Theorem 1.3. Consider a polynomial (family) $f = f_n \in \mathbb{F}_q[x_1, \ldots, x_n]$ in $\overline{\mathrm{VNP}}_\varepsilon$ of degree d, which is approximated by $g \in \mathbb{F}_q[\varepsilon, x_1, \ldots, x_n]$ as per Definition 1.2. Let the $\overline{\mathrm{VNP}}_\varepsilon$ size parameters of g be (s, s), where $s := \mathrm{poly}(n)$ and $d := \deg_{\boldsymbol{x}}(g) \leq \mathrm{poly}(s)$. The size of the verifier circuit h from Definition 1.2 is bounded by s, hence the degree $D := \deg_\varepsilon(h) \leq 2^s$ (as, w.l.o.g., h has multiplication-fanin two).

Using Lemma 4.1 on g, followed by applying Lemma 4.2, gives a #P/poly function ϕ_g which computes the encoding of coefficients of g. The coefficient of a monomial $\boldsymbol{x}^{\boldsymbol{e}}$ in f is the coefficient of $\varepsilon^M \cdot \boldsymbol{x}^{\boldsymbol{e}}$ in the approximating polynomial g. Observe that if

$$f = \sum_{\boldsymbol{e} \in \{0, \ldots, d\}^n} c_{\boldsymbol{e}} \cdot \boldsymbol{x}^{\boldsymbol{e}}, \tag{2}$$

then $\langle c_{\boldsymbol{e}} \rangle = \phi_g(M, e_1, \ldots, e_n)$. From the definition of $\overline{\mathrm{VNP}}_\varepsilon$, we know that $d, \log(M) \leq \mathrm{poly}(n)$. So, using Valiant's criterion (Proposition 2.1) we conclude that f is in VNP. □

4.1 Application: Deborder Factors

Motivated from the discussion in Section 1.2, we formally define the presentable class $\overline{\mathrm{VP}}_\varepsilon$ below.

Definition 4.3 (Presentable $\overline{\mathrm{VP}}$). The presentable border class $\overline{\mathrm{VP}}_\varepsilon$ is defined as the set of polynomials $f \in \mathbb{F}[x_1, \ldots, x_n]$ such that there is an approximating polynomial $g \in \mathbb{F}[\varepsilon][x_1, \ldots, x_n]$ satisfying

$$g(\boldsymbol{x}, \varepsilon) = \varepsilon^M f(\boldsymbol{x}) + \varepsilon^{M+1} Q(\boldsymbol{x}, \varepsilon),$$

for some $Q \in \mathbb{F}[\varepsilon][x_1, \ldots, x_n]$ and $M \in \mathbb{N}$. Moreover, $\mathrm{size}_{\mathbb{F}}(g)$ and $\deg_{\boldsymbol{x}}(g)$ is bounded by $\mathrm{poly}(n)$.

Although, g has a small size circuit, we emphasise that the degree of ε-polynomials in g is unrestricted. Further, it is apparent from the definitions that $\mathrm{VP} \subseteq \overline{\mathrm{VP}}_\varepsilon \subseteq \overline{\mathrm{VNP}}_\varepsilon$. A factor is called *separable* when it is irreducible and has multiplicity coprime to the characteristic of the field. Bürgisser proved in [14, Theorem 1.3], that the class $\overline{\mathrm{VP}}_\varepsilon$ contains all the low-degree separable factors of circuits of small size. Note that, over the field of characteristic zero the result will hold for all the low-degree factors. We state it formally in the following lemma.

Lemma 4.4. *Let $q := p^a$ and e be a positive integer coprime to p. Consider a polynomial (family) $f \in \mathbb{F}_q[x_1, \ldots, x_n]$ satisfying $f = u^e v$, where u is irreducible and coprime to v, such that $\mathrm{size}(f)$ and $\deg(u)$ is at most $s := \mathrm{poly}(n, \log q)$. Then we have u in $\overline{\mathrm{VP}}_\varepsilon$.*

Remark. We make a few observations.

(1) In case $f = u^e$, Kaltofen [39] showed that u is VP.
(2) Bürgisser [14] proved that the low-degree factor u is in $\overline{\mathrm{VP}}$. Moreover, he remarked that, in his proof, the required polynomials in $\mathbb{F}[\varepsilon]$ do have small circuit-complexity (refer the remark following [14, Definition 2.1]). For completeness, we give the proof for $u \in \overline{\mathrm{VP}}_\varepsilon$ in the full version of our paper [4].

As an application of the debordering result over finite fields in Theorem 1.3, we prove that the low-degree separable factors of small size circuits are explicit.

Corollary 1.5 (Formally restated). *Let $q := p^a$ and e be a positive integer coprime to p. Consider a family of polynomial $f \in \mathbb{F}_q[x_1, \ldots, x_n]$ and its irreducible factor u satisfying $f = u^e v$, u coprime to v, such that $\mathrm{size}(f)$ and $\deg(u)$ is $\mathrm{poly}(n, \log q)$. Then, the polynomial (family) u is in* VNP.

Proof. We learn from Lemma 4.4 that the polynomial family $u \in \overline{\mathrm{VP}}_\varepsilon$. Moreover, $\overline{\mathrm{VP}}_\varepsilon$ is contained in $\overline{\mathrm{VNP}}_\varepsilon$ by definition. As over $\mathbb{F}_q$, Theorem 1.3 proves $\overline{\mathrm{VNP}}_\varepsilon = \mathrm{VNP}$, hence $u \in \mathrm{VNP}$. □

5 VNP IS FACTOR CLOSED

In a pioneering work, Valiant [78], defined VNP as a class of polynomials which can be expressed as hypercube sum of a VP circuit (Definition 1.1). In a subsequent work [79], Valiant showed that VNP agrees with many fundamental closure properties, making it the commonly accepted definition of *explicit* polynomials in Algebraic Complexity Theory. Some of these properties are crucially required in our proofs and discussed in the following lemma.

Lemma 5.1 (VNP closure properties). *For all $i \in [t]$, let $f_i \in \mathbb{F}[x_1, \ldots, x_n, y_1, \ldots, y_m]$ be polynomials in* VNP *over $\mathbb{F}$, where t is at most $\mathrm{poly}(n, m)$. Then the following closure properties hold:*

(1) Addition and Multiplication: *Let $f_+ := \sum_{i \in [t]} f_i$, and $f_\times := \prod_{i \in [t]} f_i$. Then f_+ and $f_\times$ are in* VNP.
(2) Coefficient Extraction: *For all $i \in [t]$, let $f_i = \sum_{\boldsymbol{e}} c_{\boldsymbol{e}}(\boldsymbol{x}) \cdot \boldsymbol{y}^{\boldsymbol{e}}$. Then for all exponent vectors $\boldsymbol{e}$, the coefficient $c_{\boldsymbol{e}}$ is also a polynomial in* VNP.
(3) Composition: *Let g be a t-variate polynomial in* VNP. *Then $g(f_1, \ldots, f_t)$ is in* VNP.

For completeness we give the proof of the lemma in the full version of our paper [4]. Meanwhile we state the three technical lemmas that help us prove Theorem 1.6, specifically for the case of polynomial factoring in small characteristic fields. The first lemma is our main contribution that handles the 'pure' inseparable case of factoring.

Lemma 5.2 (Prime power). *Let $f \in \mathbb{F}_q[x_1, \ldots, x_n]$ be a polynomial in* VNP. *If there is a polynomial u and a positive integer i such that $f = u^{p^i}$, then the factor u is in* VNP.

Lemma 5.3 (Coprime factors). *Let $f \in \mathbb{F}_q[x_1, \ldots, x_n]$ be a polynomial in* VNP. *If there are co-prime polynomials u and v such that $f = u \cdot v$, then the factor u is in* VNP.

We defer the proof of the above fundamental lemmas to the subsequent two sub-sections. For now, we use them to prove an essential lemma that deals with the 'radical' computation in VNP.

Lemma 5.4 (Any power). *Let $f \in \mathbb{F}_q[x_1, \ldots, x_n]$ be a polynomial in* VNP. *If there is a polynomial u and an arbitrary positive integer e such that $f = u^e$, then the factor u is in* VNP.

Proof. Let $e := p^i \cdot \widehat{e}$, and $u_1 := u^{p^i}$, such that p does not divide $\widehat{e}$. Note that, when $\widehat{e} = 1$ then Lemma 5.2 finishes the proof. When $\widehat{e} > 1$, we associate a polynomial $\widehat{f}$ with a new variable z as follows:

$$\begin{aligned}\widehat{f} &:= z^{\widehat{e}} - f \;=\; z^{\widehat{e}} - u_1^{\widehat{e}} \\ &= (z - u_1) \cdot \left(z^{\widehat{e}-1} + z^{\widehat{e}-2}u_1 + \cdots + u_1^{\widehat{e}-1}\right) \\ &=: u_2(z) \cdot u_3(z) \ .\end{aligned}$$

For contradiction sake, assume that u_2 and u_3 share a factor, and hence are not co-prime. This implies that u_1 must be a root of u_3, which gives $u_3(u_1) = \widehat{e} \cdot u_1^{\widehat{e}-1} = 0$. However, since $\widehat{e} > 1$ and u_1 is non-zero, it follows that the characteristic p divides $\widehat{e}$, which contradicts our choice of $\widehat{e}$.

Observe that $z^{\widehat{e}}$ is trivially in VNP, hence we obtain that $\widehat{f}$ is in VNP. Since u_2 and u_3 are co-prime, we invoke Lemma 5.3 to shows that u_2 is in VNP, and therefore u_1 is in VNP. We finish the proof by using Lemma 5.2 on u_1 to finally prove that u is in VNP. □

With all the essential ingredients in place, we are now ready to prove the second main result of our paper. We will restate Theorem 1.6 formally, which proves the closure of VNP under factoring over all fields.

Theorem 1.6 (Formally restated). *Let $\mathbb{F}$ be a field of any characteristic. Consider a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ in the class* VNP *and let u be its arbitrary factor. Then, we have u in* VNP.

Proof. Over fields of characteristic zero, it was proved in [21, Theorem 2.8] that u is in VNP. Here we consider the hitherto unsolved case of small prime characteristic. In particular, when $\mathbb{F} = \mathbb{F}_q$, where $q =: p^a$ for some prime $p < \deg(f)$.

Pick the largest integer $e \geq 1$ and the polynomial v over $\mathbb{F}_q$ satisfying $f =: u^e v$. If $v = 1$, then Lemma 5.4 proves that u is in VNP.

If u and v are co-prime, then we conclude using Lemma 5.3 and Lemma 5.4.

In the last case, there exists an irreducible polynomial $w \in \mathbb{F}_q[x_1, \ldots, x_n]$ that divides both u and v. Consider $u_1 := w^{e'}$ and $v_1 := (f/u_1)$ such that u_1, v_1 are coprime factors of f. Again, using Lemma 5.3 and Lemma 5.4 we get that w is in VNP. Repeat this for all the irreducible factors of u, and use the fact that VNP is closed under multiplication (Lemma 5.1); this concludes the proof of u being in VNP. □

5.1 Factoring Prime Powers or Valiant's Converse

To prove Lemma 5.2, we show that the coefficients of the factor polynomial u can be computed effectively, and thus use Valiant's criterion to prove the claim. We will argue that coefficients of u can be obtained from the coefficient function of f. Therefore, it would suffice to design an effectively computable coefficient function for f, give that it is in VNP. To that effect, we prove the *converse* of Valiant's criterion, over finite fields.

Lemma 5.5 (Converse of Valiant's criterion). *Let $f = \sum_{\boldsymbol{e}} c_{\boldsymbol{e}} \cdot \boldsymbol{x}^{\boldsymbol{e}}$ be a polynomial in* VNP *over $\mathbb{F}_q$. Then, there exists a function ϕ_f in* #P/poly *such that for all $\boldsymbol{e}$, $\phi_f(\langle \boldsymbol{e} \rangle) = \langle c_{\boldsymbol{e}} \rangle$.*

Proof. Let $D := \deg(f)$ and the VNP size parameters of f be (s, s) where $s := \mathrm{poly}(n, \log q)$. Using the exponential-interpolation in Lemma 4.1, with $D = \mathrm{poly}(s)$, we can prove that each coefficient $c_{\boldsymbol{e}}$ of f is a hypercube-sum of small-circuit evaluations, with parameters $(\mathrm{poly}(s), \mathrm{poly}(s))$ [5]. That is, there is a polynomial $t_{\boldsymbol{e}}$ over a finite field extension $\mathbb{F}_{q'}$, $q' \leq \mathrm{poly}(s)$, such that

$$c_{\boldsymbol{e}} = \sum_{\boldsymbol{b} \in \{0,1\}^{\ell}} t_{\boldsymbol{e}}(b_1, \ldots, b_\ell),$$

where ℓ and $\mathrm{size}(t_{\boldsymbol{e}})$ are at most $\mathrm{poly}(s)$. Next, moving to the boolean world, Lemma 4.2 shows that such an algebraic representation can be transformed to obtain the coefficient function $\phi_f \in$ #P/poly such that $\phi_f(\boldsymbol{e}) = \langle c_{\boldsymbol{e}} \rangle$. □

Recall that over finite fields, and for our purposes, we work with a weaker version where coefficient function is in $\#_p$P/poly instead. Refer to the remark following Proposition 2.1.

As mentioned earlier, with the coefficient function of f in place, we need a way to map the coefficients of f to u. Following is a well-known claim from Algebra, that will help us map the coefficients.

Claim 5.6 (Frobenius Homomorphism). *Let* R *be a commutative ring of characteristic p. Define a map $\rho : \mathrm{R} \to \mathrm{R}$ as $\rho(u) = u^{p^i}$. Then, ρ is a ring homomorphism. Moreover, when* R *is a finite field $\mathbb{F}_q$, then ρ is an automorphism that fixes $\mathbb{F}_{p^i}$.*

We now have all the necessary tools needed to prove the lemma.

Proof of Lemma 5.2. Given that $f = u^{p^i}$, suppose $u =: \sum_{\boldsymbol{a} \in L} c_{\boldsymbol{a}} \boldsymbol{x}^{\boldsymbol{a}}$, where the *support* L represents the set of exponent vectors associated to u. Essentially, Claim 5.6 allows us to distribute the prime power over addition as follows:

$$f \;=\; u^{p^i} \;=\; \left(\sum_{\boldsymbol{a} \in L} c_{\boldsymbol{a}} \cdot \boldsymbol{x}^{\boldsymbol{a}}\right)^{p^i} \;=\; \sum_{\boldsymbol{a} \in L} (c_{\boldsymbol{a}})^{p^i} \, \boldsymbol{x}^{p^i \cdot \boldsymbol{a}} \ .$$

The last expression above clearly associates the coefficients of $\boldsymbol{x}^{p^i \cdot \boldsymbol{a}}$ in f to coefficients of $\boldsymbol{x}^{\boldsymbol{a}}$ in u. Since f is in VNP, Lemma 5.5 guarantees a #P/poly function ϕ_f such that the following congruence, in the finite field $\mathbb{F}_q$, is true for all $\boldsymbol{a} \in L$:

[5] The same conclusion can be made from VNP closure properties stated in Lemma 5.1.

$$\left(\phi_f\left(p^i\cdot \boldsymbol{a}\right)\right)^{1/p^i} = \phi_f(p^i\cdot\boldsymbol{a})^{q/p^i} = \phi_f(p^i\cdot\boldsymbol{a})^{p^{a-i}}$$
$$=: \phi_u(\boldsymbol{a}) = \langle c_{\boldsymbol{a}}\rangle \ .$$

Since #P/poly functions are closed under repeated-squaring, we conclude that $\phi_u \in$ #P/poly. Invoking Proposition 2.1 on ϕ_u proves that the factor $u \in$ VNP. □

5.2 Factoring Co-prime Factors

The proof of Lemma 5.3 adheres to the conventional template of factoring, pioneered by Kaltofen, using Hensel's lifting lemma. We will follow the presentation of [45, 74, 77]. It commences with a series of preprocessing procedures that brings the polynomial in the right setup to invoke the lifting lemma, which uniquely gives the factor. We will elucidate all the steps, and along the way analyse the VNP size parameters to ultimately conclude the proof.

Transformation to monic polynomial. Let $\alpha \in \mathbb{F}_q^n$ such that $\alpha := (\alpha_1, \dots, \alpha_n)$. Define a homogeneous *shift* map $\tau_\alpha : \mathbb{F}_q[x_1, \dots, x_n] \to \mathbb{F}_q[x, x_1, \dots, x_n]$ such that for all $i \in [n]$, it maps $x_i \mapsto x_i + \alpha_i \cdot x$. Let $f_\alpha := \tau_\alpha(f)$ and observe that $\deg(f_\alpha) = \deg(f) =: d$. Isolating the coefficient $c_{\boldsymbol{e}}$ of the leading term x^d of f_α gives

$$c_{\boldsymbol{e}} =: \sum_{|\boldsymbol{e}|=d} \widehat{c}_{\boldsymbol{e}} \cdot \alpha_1^{e_1} \dots \alpha_n^{e_n}.$$

PIT lemma guarantees that with high probability, a random choice of α ensures $c_{\boldsymbol{e}}$ is a non-zero field element (refer to [73, Lemma 4.2]). Then, $f_\alpha/c_{\boldsymbol{e}}$ is a monic polynomial in x. Further, if (s, s) is the VNP size parameters of f, then the parameters for f_α are $(s, s + O(n))$. When the field is too small, to pick the right α, we can obtain it from a field extension K of degree at most poly$(\deg(f))$. Since arithmetic operations over K can be efficiently simulated in $\mathbb{F}$ (refer to [13, Proposition 4.1]), we will assume $\mathsf{K} = \mathbb{F}_q$ without loss of generality.

Multivariate to bi-variate factoring. We can reduce the problem of multivariate factoring to the bi-variate case. For notational convenience, we redefine f_α/c_e as f_α and associate a polynomial $\bar{f} \in \mathbb{F}_q[x_1, \dots, x_n][x, y]$ as follows: $\bar{f}(x, y) := f_\alpha(x, yx_1 + a_1, yx_2 + a_2, \dots, yx_n + a_n)$, where $\mathbf{a} \in \mathbb{F}_q^n$ is a point.

If f_α is monic and u_α is its monic irreducible factor, then $\bar{u} := u(x, yx_1 + a_1, \dots, yx_n + a_n)$ is a monic irreducible factor of $\bar{f}$, see [74, Lemma 3.10]. In addition to this bi-variate transformation, the scaling and shifting of variables sets up the starting point for the lifting lemma. Refer to [23, Section 2.2] and [74, Section 3.5].

Claim 5.7 (Initialize Hensel lifting). *Let $f = u \cdot v$ be such that u, v are co-prime polynomials. Then the associated univariate factors $\bar{u}(x, 0)$ and $\bar{v}(x, 0)$ of $\bar{f}(x, 0)$ are co-prime.*

Note that, the factor u can be recovered easily from $\bar{u}$ by performing an inverse linear-transformation of the coordinate shift. Further, the polynomial $\bar{f}(x, y)$ remains monic in x and is in VNP with size parameters $(s, s + O(n))$.

Hensel's Lifting. Let us re-assign $f = \bar{f}$ for notational simplicity. Recall that $f(x, y)$ is monic in x, therefore $f_0 := f(x, 0) \in \mathbb{F}_q[x]$ is a univariate polynomial of degree d. Since f_0 can have at most d factors, $u_0 := u(x, 0)$ and $v_0 := v(x, 0)$ are in VNP with parameters $(1, O(d))$. We will use the following ever-famous Hensel's Lifting lemma from number theory to lift the roots uniquely (mod y). For a detailed discussion on the specific monic version of the Lifting lemma required for our proof, we encourage the readers to refer [45, Lemma 3.4]. For the rest of the section we assume $\mathbb{K} := \mathbb{F}_q[x_1, \dots, x_n]$ as the base ring of the bivariate polynomials in x, y.

Lemma 5.8 (Monic Hensel's Lifting). *Let $f = u \cdot v \in \mathbb{K}[x, y]$ be such that u, v are co-prime, and u is monic in x. Additionally, we are given $u_0 \equiv u \bmod y$ and $v_o \equiv v \bmod y$ such that $a_0u_0 + b_0v_0 \equiv 1 \bmod y$. Then for all natural numbers $k \geq 1$ there exist $u_k, v_k, a_k, b_k \in \mathbb{K}[x, y]$ satisfying the following:*

1. *$u_k \equiv u_{k-1} \bmod y^{2^{k-1}}$ and $v_k \equiv v_{k-1} \bmod y^{2^{k-1}}$.*
2. *$f \equiv u_k \cdot v_k \bmod y^{2^k}$ such that $a_ku_k + b_kv_k \equiv 1 \bmod y^{2^k}$ and u_k is* monic *in x.*
3. *$u_k \equiv u \bmod y^{2^k}$ and $v_k \equiv v \bmod y^{2^k}$.*

Moreover, for every k, the lifted factors u_k and v_k are unique *polynomials* mody^{2^k}.

Hensel's Lifting is a technical, but a very powerful tool which gives explicit formulas for the lifted factors. Its basic idea is to take the error of the previous step and *feed it back* to the next step. Consider the difference polynomial $m_k := f - u_{k-1}v_{k-1}$. Then the polynomials $\bar{u}_k := u_{k-1} + b_{k-1}m_k$ and $\bar{v}_k := v_{k-1} + a_{k-1}m_k$ are valid lifts of the factors u and v. However, to obtain monic, and therefore unique lifts, we need some correction. Let $q_k, r_k \in \mathbb{K}[x, y]$ be such that

$$(\bar{u}_k - u_{k-1}) \ =: \ y^{2^{k-1}} \cdot (q_ku_{k-1} + r_k)\,,$$

where $\deg_x(r_k) \leq \deg_x(u_{k-1})$. The existence of these polynomials is guaranteed by Euclid's division algorithm. Then the unique, and monic, lifts are defined as follows:

$$u_k := u_{k-1} + y^{2^{k-1}} r_k \tag{3}$$
$$v_k := \bar{v}_k \left(1 + y^{2^{k-1}} q_k\right) \ . \tag{4}$$

It is easy to verify that they are the valid lifts as per Lemma 5.8. Refer [45, Lemma 3.4] for rigorous calculations. In addition, let $w_k := a_{k-1}u_k + b_{k-1}v_k$, then the lifted factors remain (pseudo-)co-prime $(\bmod\ y^{2^k})$ with Bézout identity holding using the following polynomials:

$$a_k := a_{k-1}(1 - w_k)$$
$$b_k := b_{k-1}(1 - w_k)\,.$$

Size analysis. We choose an integer $t \geq \log(\deg_y(u)) + 1$ and repeatedly use the Lifting lemma t times to obtain the factor $u_t \equiv u \bmod y^{2^t}$. Since the lifted factors are unique, u can be obtained from u_t by truncating it to $\deg_y(u)$. Given that $f \in$ VNP, the factor $u \in$ VNP can be proved using the following technical lemma. It proves that given the coefficients of polynomial f in variables $x_1, \dots, x_n$, there is a small circuit which computes the lifted factor u.

Lemma 5.9 (Hensel in circuits). *Let $f = u \cdot v \in \mathbb{K}[x, y]$ be a degree d polynomial such that u, v are co-prime and u is monic in x. The polynomials u_0, v_0, a_0, b_0 are defined as before. Let L be the set of exponent vectors of f such that $f =: \sum_{\boldsymbol{e}_i \in L} c_{\boldsymbol{e}_i}(x_1, \dots, x_n) \cdot x^{e_{i1}} y^{e_{i2}}$.*

Given the coefficients $c_{\boldsymbol{e}_1}, \ldots, c_{\boldsymbol{e}_{|L|}}$ *as input, there exists a circuit* $C_u^{(t)}$ *over* $\mathbb{F}_q$ *which computes* $\mathrm{Hom}_{\leq d}(u_t)$[6]*. Further, there is a constant* $\beta \geq 2$ *such that the size of the circuit* $C_u^{(t)}$ *is at most* $\mathrm{poly}(d, \beta^t)$*, and intermediate degrees at most* $(d\beta^t)$*.*

Proof. Given all the coefficients of the polynomial f, observe that we can construct a sub-circuit C_f of size $s_f := \mathrm{poly}(d)$ that computes f. Then, the proof is an easy consequence of the following inductive analysis on t.

The base case is easy to analyse. Let $C_u^{(t-1)}, C_v^{(t-1)}, C_a^{(t-1)}$, and $C_b^{(t-1)}$ be the circuits that compute $u_{t-1}, v_{t-1}, a_{t-1}$ and b_{t-1} respectively, as described in Hensel's lifting Lemma 5.8. Let the size of all the circuits be at most $s_{t-1} := \mathrm{poly}(d, \beta^{t-1})$. Together with C_f, the difference polynomial m_k can be easily computed in size $s_f + O(s_{t-1})$ [7]. Then observe that $\mathrm{size}(\bar{u}_t)$ and $\mathrm{size}(\bar{v}_t)$ is at most $s_f + O(s_{t-1})$. To facilitate the lifting process, the quotient q_k and remainder r_k can be computed with additional $\mathrm{poly}(d)$ size (refer [45, Lemma 2.8] and [81, Lemma 9.6]). Using these as sub-circuits, we obtain C_u^t and C_v^t with additional constant number of gates from Equations 3 and 4. Overall, the size of the lifted polynomials grows by a constant factor and, hence, the overall size of both the circuits is at most $s_t := s_f + O(s_{t-1}) + \mathrm{poly}(d) + O(\beta) \leq \mathrm{poly}(d, \beta^t)$. Almost the same argument works for circuits $C_a^{(t)}$ and $C_b^{(t)}$ computing a_t and b_t.

Lastly, we homogenize C_u^t, to obtain the desired circuit which computes $\mathrm{Hom}_{\leq d}(u_t)$. The degree with respect to the lifting variable y is at most β^t due to constant growth in each iteration, moreover, with respect to x it is at most d due to the homogenization. Hence, the degree claim follows. □

We are now ready to give the complete proof of the following Lemma 5.3.

Lemma 5.3 (restated). *Let* $f \in \mathbb{F}_q[x_1, \ldots, x_n]$ *be a polynomial in* VNP*. If there are co-prime polynomials* u *and* v *such that* $f = u \cdot v$*, then the factor* u *is in* VNP*.*

Proof. Assume that $f \in \mathbb{K}[x, y]$ after all the necessary invertible transformations discussed earlier in the section to apply Lemma 5.8. Support L be the set of exponent vectors of f such that $f =: \sum_{\boldsymbol{e}_i \in L} c_{\boldsymbol{e}_i}(x_1, \ldots, x_n) \cdot x^{e_{i1}} y^{e_{i2}}$.

Using Lemma 5.9 with $t \geq \log(\deg(f)) + 1$ gives a circuit $C_u^{(t)}$ that take the coefficients of f as input and outputs a circuit for the factor u. Moreover, the size of the circuit is at most $\mathrm{poly}(\deg(f))$ and degree is at most $O(\deg(f))$.

Since $f \in$ VNP, Lemma 5.1(2) shows that the coefficients $c_{\boldsymbol{e}_i} \in$ VNP. Moreover, Lemma 5.1(3) will prove that $C_u^{(t)}$ composed with VNP polynomials, remains in VNP. Therefore, the factor u is in VNP. □

6 CONCLUSION

Motivated by the need of an expressive model of approximation, in this work, we defined *presentable* border classes $\overline{\mathsf{VP}}_\varepsilon$ and $\overline{\mathsf{VNP}}_\varepsilon$. We proved that $\overline{\mathsf{VNP}}_\varepsilon$ is contained in VNP, over finite fields. The question whether $\overline{\mathsf{VNP}}_\varepsilon$ is contained in VNP, remains open over $\mathbb{Q}$; due to the possibility of *doubly*-exponentially large integers appearing.

As an application of our debordering result, we advance partially towards proving the factor conjecture [13, Conjecture 8.3], by showing that low-degree 'separable' factors of small size circuits are explicit. This still does not rule out the possibility: Could the Permanent polynomial be a factor of a small circuit of exponential degree?

Our debordering technique, of exponential interpolation, further proved that over all finite fields, VNP is closed under factoring, and thus resolves Bürgisser's conjecture [13, Conjecture 2.1].

Whitebox PIT.. Our newly introduced *presentable* border classes open up a new avenue of studying Polynomial Identity Testing (PIT) in the *whitebox* setting. PIT is a fundamental problem in complexity theory, that decides the zeroness of the given polynomial (refer [67, 68] and also [73, Chapter 4]). It is studied under two well known settings: *Blackbox* and *Whitebox*. The former allows only evaluations, while the latter allows to look at the inner structure of the model. PIT on border classes, naturally extends to testing the zeroness of a polynomial, given its approximating polynomial. Concretely, let g approximate a non-zero polynomial $f \in \overline{\mathsf{VP}}$, then there exists an evaluation point α such that $g(\alpha, \varepsilon)$ is not a multiple of ε. We emphasise that mere non-zeroness of $g(\alpha, \varepsilon)$ does not guarantee non-zeroness of f. For a comprehensive discussion and motivations of blackbox border PIT, refer [23, 30].

The arbitrarily large complexity of ε-polynomial in g, makes the whitebox testing in border classes a meaningless problem; as the input cannot be presented. But now the *presentable* border classes such as $\overline{\mathsf{VP}}_\varepsilon$ constrain the ε-polynomials, and therefore we make the whitebox setting interesting. It is worthwhile to investigate whitebox PIT on presentable border classes; for instance, study constant depth circuits to begin with.

ACKNOWLEDGMENTS

The authors would like to thank Ramprasad Saptharishi, and Sylvain Perifel for the email conversations on Valiant's criterion. The genesis of the work was possible thanks to the conducive atmosphere of the Workshop on Algebraic Complexity Theory 2023 organised by Christian Ikenmeyer in University of Warwick, UK. N.S. thanks the Workshop on Recent Trends in Computer Algebra 2023 for giving the opportunity to discuss the details of [69] in IHP, Paris; and the funding support from DST-SERB (CRG/2020/000045 + JCB/2022/57), and N. Rama Rao Chair (2019–). C.S.B. thanks Microsoft Research Lab India, and IARCS-ACM for supporting the travel expenses.

REFERENCES

[1] Manindra Agrawal, Sumanta Ghosh, and Nitin Saxena. 2019. Bootstrapping variables in algebraic circuits. *Proc. Natl. Acad. Sci. USA* 116, 17 (2019), 8107–8118. https://doi.org/10.1073/pnas.1901272116

[2] Eric Allender and Fengming Wang. 2016. On the power of algebraic branching programs of width two. *Comput. Complexity* 25, 1 (2016), 217–253. https://doi.org/10.1007/s00037-015-0114-7

[3] Michael Ben-Or and Richard Cleve. 1992. Computing algebraic formulas using a constant number of registers. *SIAM J. Comput.* 21, 1 (1992), 54–58. https://doi.org/10.1137/0221006

[6] This is the sum of the homogeneous parts of u_t up to degree d.

[7] For notations, refer to the discussion proceeding Lemma 5.8.

[4] C.S. Bhargav, Prateek Dwivedi, and Nitin Saxena. 2024. Learning the Coefficients: A Presentable Version of Border Complexity and Applications to Circuit Factoring. (2024). https://cse.iitk.ac.in/users/nitin/papers/PresentableVNP.pdf

[5] Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. 2020. Deterministic factorization of sparse polynomials with bounded individual degree. *J. ACM* 67, 2 (2020), Art. 8, 28. https://doi.org/10.1145/3365667

[6] D. Bini. 1980. Relations between exact and approximate bilinear algorithms. Applications. *Calcolo. A Quarterly on Numerical Analysis and Theory of Computation* 17, 1 (1980), 87–97. https://doi.org/10.1007/BF02575865

[7] Dario Bini, Milvio Capovani, Francesco Romani, and Grazia Lotti. 1979. $O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication. *Inform. Process. Lett.* 8, 5 (1979), 234–235. https://doi.org/10.1016/0020-0190(79)90113-3

[8] Pranav Bisht and Nitin Saxena. 2021. Blackbox identity testing for sum of special ROABPs and its border class. *Comput. Complexity* 30, 1 (2021), Paper No. 8, 48. https://doi.org/10.1007/s00037-021-00209-y

[9] Markus Bläser, Christian Ikenmeyer, Meena Mahajan, Anurag Pandey, and Nitin Saurabh. 2020. Algebraic Branching Programs, Border Complexity, and Tangent Spaces. In *Proceedings of the 35th Annual Computational Complexity Conference (CCC 2020)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 21:1–21:24. https://doi.org/10.4230/LIPICS.CCC.2020.21

[10] Markus Bläser and Christian Ikenmeyer. 2018. Introduction to geometric complexity theory. (2018). https://www.dcs.warwick.ac.uk/~u2270030/teaching_sb/summer17/introtogct/gct.pdf Lecture Notes.

[11] Karl Bringmann, Christian Ikenmeyer, and Jeroen Zuiddam. 2018. On algebraic branching programs of small width. *J. ACM* 65, 5 (2018), Art. 32, 29. https://doi.org/10.1145/3209663

[12] Peter Bürgisser, Michael Clausen, and Mohammad Amin Shokrollahi. 1997. *Algebraic complexity theory*. Grundlehren der mathematischen Wissenschaften, Vol. 315. Springer. https://doi.org/10.1007/978-3-662-03338-8

[13] Peter Bürgisser. 2000. *Completeness and reduction in algebraic complexity theory*. Algorithms and computation in mathematics, Vol. 7. Springer-Verlag. https://doi.org/10.1007/978-3-662-04179-6

[14] Peter Bürgisser. 2004. The complexity of factors of multivariate polynomials. *Foundations of Computational Mathematics* 4, 4 (2004), 369–396. https://doi.org/10.1007/s10208-002-0059-5 Preliminary version in the *42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2001)*.

[15] Peter Bürgisser. 2020. Correction to: The complexity of factors of multivariate polynomials. *Foundations of Computational Mathematics* 20, 6 (2020), 1667–1668. https://doi.org/10.1007/s10208-020-09477-6

[16] Prasad Chaugule and Nutan Limaye. 2022. On the closures of monotone algebraic classes and variants of the determinant. In *LATIN 2022: theoretical informatics*. Lecture Notes in Comput. Sci., Vol. 13568. Springer, 610–625. https://doi.org/10.1007/978-3-031-20624-5_37

[17] Xi Chen, Neeraj Kayal, and Avi Wigderson. 2010. Partial derivatives in arithmetic complexity and beyond. *Found. Trends Theor. Comput. Sci.* 6, 1-2 (2010). https://doi.org/10.1561/0400000043

[18] Mahdi Cheraghchi, Elena Grigorescu, Brendan Juba, Karl Wimmer, and Ning Xie. 2018. $AC^0 \circ MOD_2$ lower bounds for the Boolean inner product. *J. Comput. System Sci.* 97 (2018), 45–59. https://doi.org/10.1016/j.jcss.2018.04.006

[19] Benny Chor and Ronald L. Rivest. 1988. A knapsack-type public key cryptosystem based on arithmetic in finite fields. *IEEE Trans. Inform. Theory* 34, 5, part 1 (1988), 901–909. https://doi.org/10.1109/18.21214

[20] Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. 2019. Closure of VP under taking factors: a short and simple proof. arXiv:1903.02366 [cs.CC] https://arxiv.org/abs/1903.02366

[21] Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. 2019. Closure results for polynomial factorization. *Theory of Computing. An Open Access Journal* 15 (2019), Paper No. 13, 34. https://doi.org/10.4086/toc.2019.v015a013 Preliminary version in the *33rd Annual Computational Complexity Conference (CCC 2018)*.

[22] Don Coppersmith and Shmuel Winograd. 1990. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation* 9, 3 (1990), 251–280. https://doi.org/10.1016/S0747-7171(08)80013-2

[23] Pranjal Dutta, Prateek Dwivedi, and Nitin Saxena. 2021. Demystifying the border of depth-3 algebraic circuits. In *Proceedings of the 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2021)*. IEEE, 92–103. https://doi.org/10.1109/FOCS52979.2021.00018

[24] Pranjal Dutta and Nitin Saxena. 2022. Separated borders: Exponential-gap fanin-hierarchy theorem for approximative depth-3 circuits. In *Proceedings of the 63rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2022)*. IEEE, 200–211. https://doi.org/10.1109/FOCS54457.2022.00026

[25] Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. 2022. Discovering the Roots: Uniform Closure Results for Algebraic Classes Under Factoring. *J. ACM* 69, 3 (2022), 18:1–18:39. https://doi.org/10.1145/3510359

[26] Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. 2009/10. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM J. Comput.* 39, 4 (2009/10), 1279–1293. https://doi.org/10.1137/080735850

[27] Michael Forbes. 2016. Some Concrete Questions on the Border Complexity of Polynomials. https://www.youtube.com/watch?v=1HMogQIHT6Q Talk at the Workshop on Algebraic Complexity Theory (WACT) 2016 in Tel Aviv..

[28] Michael Forbes and Amir Shpilka. 2015. Complexity Theory Column 88: Challenges in Polynomial Factorization1. *SIGACT News* 46, 4 (Dec 2015), 32–49. https://doi.org/10.1145/2852040.2852051

[29] Michael A. Forbes. 2014. *Polynomial identity testing of read-once oblivious algebraic branching programs*. Ph. D. Dissertation. Massachusetts Institute of Technology, Cambridge, MA, USA. https://hdl.handle.net/1721.1/89843

[30] Michael A. Forbes and Amir Shpilka. 2018. A PSPACE construction of a hitting set for the closure of small algebraic circuits. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC 2018)*. ACM. https://doi.org/10.1145/3188745.3188792

[31] Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. 2021. Proof complexity lower bounds from algebraic circuit complexity. *Theory Comput.* 17 (2021), Paper No. 10, 88. https://doi.org/10.4086/toc.2021.v017a010

[32] Bruno Grenet. 2016. Bounded-degree factors of lacunary multivariate polynomials. *J. Symbolic Comput.* 75 (2016), 171–192. https://doi.org/10.1016/j.jsc.2015.11.013

[33] Joshua A. Grochow, Ketan D. Mulmuley, and Youming Qiao. 2016. Boundaries of VP and VNP. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP 2016) (LIPIcs, Vol. 55)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 34:1–34:14. https://doi.org/10.4230/LIPICS.ICALP.2016.34

[34] Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. 2022. Derandomization from algebraic hardness. *SIAM J. Comput.* 51, 2 (2022), 315–335. https://doi.org/10.1137/20M1347395

[35] Venkatesan Guruswami and Madhu Sudan. 1999. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Trans. Inform. Theory* 45, 6 (1999), 1757–1767. https://doi.org/10.1109/18.782097

[36] Maurice J. Jansen. 2011. Extracting Roots of Arithmetic Circuits by Adapting Numerical Methods. In *Innovations in Computer Science - ICS*. Tsinghua University Press, 87–100. http://conference.iiis.tsinghua.edu.cn/ICS2011/content/papers/4.html

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

[38] Erich Kaltofen. 1985. Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization. *SIAM J. Comput.* 14, 2 (1985), 469–489. https://doi.org/10.1137/0214035

[39] Erich Kaltofen. 1987. Single-Factor Hensel Lifting and Its Application to the Straight-Line Complexity of Certain Polynomials. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC 1987)*. Association for Computing Machinery, 443–452. https://doi.org/10.1145/28395.28443

[40] Erich Kaltofen. 1989. Factorization of Polynomials Given by Straight-Line Programs. *Adv. Comput. Res.* 5 (1989), 375–412. https://users.cs.duke.edu/~elk27/bibliography/89/Ka89_slpfac.pdf

[41] Erich Kaltofen and Pascal Koiran. 2008. Expressing a fraction of two determinants as a determinant. In *Proceedings of the 2008 International Symposium on Symbolic and Algebraic Computation (ISSAC 2008)*. ACM, 141–146. https://doi.org/10.1145/1390768.1390790

[42] Erich Kaltofen and Barry M. Trager. 1990. Computing with polynomials given by black boxes for their evaluations: greatest common divisors, factorization, separation of numerators and denominators. *J. Symbolic Comput.* 9, 3 (1990), 301–320. https://doi.org/10.1016/S0747-7171(08)80015-6

[43] Erich L. Kaltofen. 1986. Uniform Closure Properties of P-Computable Functions. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC 1986)*. ACM, 330–337. https://doi.org/10.1145/12130.12163

[44] Pascal Koiran and Sylvain Perifel. 2011. Interpolation in Valiant's theory. *Comput. Complexity* 20, 1 (2011), 1–20. https://doi.org/10.1007/s00037-011-0002-8

[45] Swastik Kopparty, Shubhangi Saraf, and Amir Shpilka. 2015. Equivalence of polynomial identity testing and polynomial factorization. *Computational Complexity* 24, 2 (2015), 295–331. https://doi.org/10.1007/s00037-015-0102-y

[46] Mrinal Kumar. 2020. On the Power of Border of Depth-3 Arithmetic Circuits. *ACM Trans. Comput. Theory* 12, 1 (2020), 5:1–5:8. https://doi.org/10.1145/3371506

[47] Mrinal Kumar and Ramprasad Saptharishi. 2019. Hardness-Randomness Tradeoffs for Algebraic Computation. *Bull. EATCS* 129 (2019). http://bulletin.eatcs.org/index.php/beatcs/article/view/591/599

[48] Mrinal Kumar, Ramprasad Saptharishi, and Anamay Tengse. 2019. Near-optimal bootstrapping of hitting sets for algebraic circuits. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 639–646. https://doi.org/10.1137/1.9781611975482.40

[49] J. M. Landsberg. 2017. *Geometry and Complexity Theory*. Cambridge University Press. https://doi.org/10.1017/9781108183192

[50] Joseph M. Landsberg and Giorgio Ottaviani. 2015. New lower bounds for the border rank of matrix multiplication. *Theory of Computing. An Open Access Journal* 11 (2015), 285–298. https://doi.org/10.4086/toc.2015.v011a011

[51] H. W. Lenstra, Jr. 1999. Finding small degree factors of lacunary polynomials. In *Number theory in progress, Vol. 1 (Zakopane-Kościelisko, 1997)*. de Gruyter, 267–276. https://doi.org/10.1515/9783110285581.267

[52] Meena Mahajan. 2014. *Algebraic Complexity Classes.* Springer International Publishing, 51–75. https://doi.org/10.1007/978-3-319-05446-9_4

[53] Guillaume Malod. 2003. *Polynômes et coefficients. (Polynomials and coefficients).* Ph. D. Dissertation. Claude Bernard University Lyon , France. https://tel.archives-ouvertes.fr/tel-00087399

[54] Guillaume Malod and Natacha Portier. 2008. Characterizing Valiant's algebraic complexity classes. *J. Complex.* 24, 1 (2008), 16–38. https://doi.org/10.1016/J.JCO.2006.09.006

[55] Gary L. Mullen and Daniel Panario. 2013. Introduction to finite fields : Basic properties of finite fields. In *Handbook of Finite Fields.* CRC Press, 13–31. https://doi.org/10.1201/b15006

[56] Ketan D. Mulmuley. 2011. On P vs. NP and geometric complexity theory. *J. ACM* 58, 2 (2011), Art. 5, 26. https://doi.org/10.1145/1944345.1944346

[57] Ketan D. Mulmuley. 2012. The GCT Program toward the P vs. NP Problem. *Commun. ACM* 55, 6 (2012), 98–107. https://doi.org/10.1145/2184319.2184341

[58] Ketan D. Mulmuley and Milind Sohoni. 2001. Geometric complexity theory. I. An approach to the P vs. NP and related problems. *Siam Journal On Computing* 31, 2 (2001), 496–526. https://doi.org/10.1137/S009753970038715X

[59] Ketan D. Mulmuley and Milind Sohoni. 2008. Geometric complexity theory. II. Towards explicit obstructions for embeddings among class varieties. *SIAM J. Comput.* 38, 3 (2008), 1175–1206. https://doi.org/10.1137/080718115

[60] David Mumford. 1976. *Algebraic geometry. I.* Springer-Verlag. x+186 pages. https://link.springer.com/book/9783540586579 Complex projective varieties.

[61] Rafael Oliveira. 2016. Factors of low individual degree polynomials. *Comput. Complexity* 25, 2 (2016), 507–561. https://doi.org/10.1007/s00037-016-0130-2

[62] Rafael Oliveira. 2020. Conditional lower bounds on the spectrahedral representation of explicit hyperbolicity cones. In *Proceedings of the 2020 International Symposium on Symbolic and Algebraic Computation (ISSAC 2020).* ACM, 396–401. https://doi.org/10.1145/3373207.3404010

[63] Christos H. Papadimitriou. 1994. *Computational complexity.* Addison-Wesley Publishing Company, Reading, MA. xvi+523 pages. https://dl.acm.org/doi/abs/10.5555/1074100.1074233

[64] Sylvain Périfel. 2004. *Polynômes donnés par des circuits algébriques et généralisation du modèle de Valiant.* École Normal Supérieure de Lyon, France. Master's Thesis.

[65] Kenneth W. Regan. 2002. Understanding the Mulmuley-Sohoni approach to P vs. NP. *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS* 78 (2002), 86–99. https://cse.buffalo.edu/faculty/regan/papers/pdf/Reg02MSFD.pdf

[66] Ramprasad Saptharishi. 2015. A survey of lower bounds in arithmetic circuit complexity. (2015). https://github.com/dasarpmar/lowerbounds-survey Github Survey.

[67] Nitin Saxena. 2009. Progress on polynomial identity testing. *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS* 99 (2009), 49–79. https://cse.iitk.ac.in/users/nitin/papers/pit-survey09.pdf

[68] Nitin Saxena. 2014. Progress on polynomial identity testing-II. In *Perspectives in computational complexity.* Progr. Comput. Sci. Appl. Logic, Vol. 26. Birkhäuser/Springer, 131–146. https://cse.iitk.ac.in/users/nitin/papers/pit-survey13.pdf

[69] Nitin Saxena. 2023. Closure of algebraic classes under factoring. https://www.cse.iitk.ac.in/users/nitin/talks/Sep2023-paris.pdf Talk at Recent Trends in Computer Algebra (2023) in Institut Henri Poincaré, Paris..

[70] Jayalal Sharma and Dinesh K. 2012. Advanced Complexity Theory. (2012). https://www.cse.iitm.ac.in/~jayalal/teaching/CS6840/2012/lecture04.pdf Lecture Notes.

[71] Victor Shoup. 2009. *A computational introduction to number theory and algebra* (second ed.). Cambridge University Press. xviii+580 pages. https://shoup.net/ntb/

[72] Amir Shpilka and Ilya Volkovich. 2010. On the relation between polynomial identity testing and finding variable disjoint factors. In *Automata, languages and programming. Part I.* Lecture Notes in Comput. Sci., Vol. 6198. Springer, 408–419. https://doi.org/10.1007/978-3-642-14165-2_35

[73] Amir Shpilka and Amir Yehudayoff. 2010. Arithmetic Circuits: A Survey of Recent Results and Open Questions. *Found. Trends Theor. Comput. Sci.* 5, 3–4 (2010), 207–388. https://doi.org/10.1561/0400000039

[74] Amit Sinhababu and Thomas Thierauf. 2021. Factorization of Polynomials Given by Arithmetic Branching Programs. *Comput. Complex.* 30, 2 (2021), 15. https://doi.org/10.1007/S00037-021-00215-0

[75] Volker Strassen. 1974. Polynomials with rational coefficients which are hard to compute. *Siam Journal On Computing* 3 (1974), 128–149. https://doi.org/10.1137/0203010

[76] Madhu Sudan. 1997. Decoding of Reed Solomon codes beyond the error-correction bound. *J. Complexity* 13, 1 (1997), 180–193. https://doi.org/10.1006/jcom.1997.0439

[77] Madhu Sudan. 1998. Algebra and Computation. (1998). https://people.csail.mit.edu/madhu/FT98/ Lecture Notes.

[78] Leslie G. Valiant. 1979. Completeness Classes in Algebra. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing (STOC 1979).* ACM, 249–261. https://doi.org/10.1145/800135.804419

[79] L. G. Valiant. 1982. Reducibility by algebraic projections. In *Logic and algorithmic.* Monogr. Enseign. Math., Vol. 30. Univ. Genève, Geneva, 365–380.

[80] Joachim von zur Gathen. 1984. Hensel and Newton methods in valuation rings. *Math. Comp.* 42, 166 (1984), 637–661. https://doi.org/10.2307/2007608

[81] Joachim von zur Gathen and Jürgen Gerhard. 2013. *Modern computer algebra* (third ed.). Cambridge University Press, Cambridge. xiv+795 pages. https://doi.org/10.1017/CBO9781139856065

[82] Joachim von zur Gathen and Erich L. Kaltofen. 1985. Factoring Sparse Multivariate Polynomials. *J. Comput. Syst. Sci.* 31, 2 (1985), 265–287. https://doi.org/10.1016/0022-0000(85)90044-3

Received 13-NOV-2023; accepted 2024-02-11