Learning the coefficients: A presentable version of border complexity and applications to circuit factoring

C. S. Bhargav [*] Prateek Dwivedi [†] Nitin Saxena [‡]

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 $\mathsf{P} \neq \mathsf{NP}$ 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).

[*]CSE, IIT Kanpur, bhargav@cse.iitk.ac.in
[†]ITU Copenhagen, prdw@itu.dk
[‡]CSE, IIT Kanpur, nitin@cse.iitk.ac.in

Contents

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 [Val79, Val82], the *border complexity* of a polynomial measures how efficiently it can be approximated. In this framework, a family of multivariate polynomial is computed by a non-uniform model called an *algebraic circuit* – a directed acyclic graph with internal nodes labeled by $+$ and $\times$ 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 size (the number of vertices and edges) of the graph. We denote the size of the smallest circuit (over $\mathbb{F}$) computing the polynomial f by $\text{size}_{\mathbb{F}}(f)$. Valiant [Val79] hypothesized that there are *explicit* polynomials that cannot be computed by circuits of small size. It is formalized as what we now call the $\mathsf{VP} \neq \mathsf{VNP}$ conjecture. The class VP (Valiant's P) consists of all polynomials with degree polynomial in the number of variables n (=: $\text{poly}(n)$), which can be computed by algebraic circuits of size $\text{poly}(n)$. He also defined an algebraic analogue of NP 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 $m = \text{poly}(n)$ 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 $\mathsf{VP} \subseteq \mathsf{VNP}$, and Valiant's conjecture is that the inclusion is strict. The surveys of [SY10, CKW10, Mah14, Sap15] provide an excellent overview of algebraic complexity and the current state of lower bounds. For a more extensive but slightly dated treatment, see [BCS97, Bü00].

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., [BI18, Section 2.3]). The topological notion has been extensively studied in the context of designing algorithms for matrix multiplication [Str74, BCRL79, Bin80, CW90, LO15]. However, in Valiant's framework, the simplest definition for *algebraic* approximation and border complexity was given by Bürgisser [Bü04]. 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 smallest circuit computing 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 abstract of our paper) 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. For a discussion of the different notions of approximation and their equivalence, see [Bü04, Lemma 5.6], [BIZ18, Section 2] and also [Mum76, Theorem 2.33].

As a natural extension, we can define the *approximate closure* of VP, called $\overline{\mathsf{VP}}$ as the set of poly(n)-degree polynomials whose *border size* is bounded by poly(n). Clearly, $\mathsf{VP} \subseteq \overline{\mathsf{VP}}$. In an ambitious program to resolve the $\mathsf{P} \overset{?}{=} \mathsf{NP}$ question using methods from algebraic geometry and representation theory, Mulmuley and Sohoni [MS01] strengthened Valiant's conjecture by postulating that VNP is not contained in $\overline{\mathsf{VP}}$ [1]. Their proposal (detailed further in [MS08]) was to use techniques from representation theory to prove lower bounds on border complexity. For expository references on the GCT program, see [Reg02, Mul11, Mul12, Lan17, BI18].

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

Completely independently and almost at the same time, Bürgisser [Bü04, Bü20] 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 [Kal85, Kal86, Kal87, Kal89], Kaltofen showed that over fields of characteristic zero, the class VP is closed under taking factors (also see [KT90]). In fact, if a polynomial factorizes as $f = u^e v$ with u and v co-prime, then Kaltofen [Kal87] showed that u can be computed by a circuit of size $\mathrm{poly}(e, \deg(u), \mathrm{size}(f))$. 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 $\mathrm{poly}(n)$-degree factor of a $\mathrm{poly}(n)$-size circuit of unrestricted degree is in VP. This is known as the Factor Conjecture [Bü00, Conjecture 8.3]. Bürgisser [Bü04] showed that for border complexity, the factor conjecture is indeed true – the factor u above, is in $\overline{\mathsf{VP}}$. 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 the 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 $\mathsf{VP} \stackrel{?}{=} \overline{\mathsf{VP}}$ [Bü04, Problem 4.3]. In a recent work [DDS21] asked a more general question, which they called *de-bordering*. Given a polynomial $f \in \overline{\mathcal{C}}$ in the approximate closure of a class $\mathcal{C}$, what is an upper bound on the exact (non-approximate) complexity of f? Although one might expect a class to not differ too much from its border class (a class $\mathcal{C}$ is *border-closed* if $\mathcal{C} = \overline{\mathcal{C}}$), 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 $\overline{\mathsf{VP}}$ 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 $\overline{\mathsf{VP}}_\varepsilon$ is the same as $\overline{\mathsf{VP}}$ 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 4.10).

There has previously been an attempt via 'degenerations' [GMQ16] to identify a subclass of $\overline{\mathsf{VP}}$ 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 $\overline{\mathsf{VP}}$ and *cannot* be obtained as a p-definable degeneration of VP, making our notion incomparable to the concept of degeneration as studied in [GMQ16]. We can extend our concept of presentable border to $\overline{\mathsf{VNP}}_\varepsilon$ over any field $\mathbb{F}$.

Definition 1.2 (Presentable $\overline{\mathsf{VNP}}$)**.** *The presentable border class $\overline{\mathsf{VNP}}_\varepsilon$, 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* $\mathrm{size}_{\mathbb{F}}(h)$ *all bounded by* $\mathrm{poly}(n)$*, satisfying a* hypercube-sum *expression*

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

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

As a consequence of our debordering result, we also make progress toward the aforementioned Factor Conjecture [Bü00, Conjecture 8.3]. As noted earlier, Bürgisser showed that any $\mathrm{poly}(n)$-degree factor of a $\mathrm{poly}(n)$-size circuit is in $\overline{\mathsf{VP}}$. We observe that it is in fact in $\overline{\mathsf{VP}}_\varepsilon$, and thus by Theorem 1 in VNP.

Corollary 1.3 (Debordering factors)**.** *Let* f_n *be a* n*-variate polynomial family over a finite field that has a* $\mathrm{poly}(n)$*-degree irreducible factor* u_n *of multiplicity co-prime to the characteristic of the field. If* $\mathrm{size}(f_n)$ *is* $\mathrm{poly}(n)$*, 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 [Bü00, Conjecture 2.1] asked if the class VNP is closed under factorization. Over fields of characteristic zero, Chou, Kumar and Solomon [CKS19b] showed that this is indeed true. Inspired by the proof technique of Theorem 1, in our second main result, we use similar techniques to prove that closure of VNP under factoring holds over finite fields as well.

Theorem 2 (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 [Bü00, 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. It remains open to answer [Bü00, Conjecture 2.1] over infinite fields of positive characteristic.[2]

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 his foundational work, Bürgisser [Bü04, Theorem 5.7] showed that over algebraically closed fields, the order of approximation M is at most exponential in $\overline{\text{size}}(f) := \text{size}_{\mathbb{F}[\varepsilon]}(g)$, 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 [3], interpolation seems to give a bound of the form $\text{size}(f) \leq \exp(\overline{\text{size}}(f))$.

Known debordering results. Incidentally, the known debordering results for restricted models of computation seldom use interpolation. In the workshop seminar [For16], Forbes remarked that Nisan's characterization implies the closure of ROABPs or equivalently non-commutative ABPs (see [For14, Chapter 4] for definitions and [BS21, 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 [BIM+20, CL22]. 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 [BIZ18]. Together

[2]Robert Andrews communicated to us an (as yet) unpublished proof that extends Theorem 2 to all *perfect* fields of positive characteristic.

[3]See [BIZ18, Corollary 3.10] for an example of debordering through interpolation when a related measure of approximation called 'error degree' is *polynomially* bounded.

with the results of [BOC92, AW16], this implies that width-2 algebraic branching programs are *not* border-closed!

In a similar vein, Kumar [Kum20] 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 size [CGJ$^+$18, Kum20]. A recent work of Dutta, Dwivedi and Saxena [DDS21] 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 [DS22] 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 [Val79] (also see [Bü00, Prop. 2.20]) which 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 $f = \sum_{\boldsymbol{e}} c_{\boldsymbol{e}} \boldsymbol{x}^{\boldsymbol{e}}$ be a polynomial in n variables of degree $\mathrm{poly}(n)$ over a field $\mathbb{F}$. Suppose that there exists a string function $\phi : \{0,1\}^* \mapsto \{0,1\}^*$ in $\#\mathsf{P}/\mathrm{poly}$ 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 $\#\mathsf{P}$ 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 [Sho09, Chapter 19] and [MP13]). 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 $\phi \in \#\mathsf{P}/\mathrm{poly}$, we assume $\phi \in \#_p\mathsf{P}/\mathrm{poly}$ [Bü00, Section 4.3]. Formally that means, there exists a function $\psi \in \#\mathsf{P}/\mathrm{poly}$ such that $\phi(\langle \boldsymbol{e} \rangle) = \psi(\langle \boldsymbol{e} \rangle) \bmod p$ [4]. We omit this subtlety wherever it is clear from the context.

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 $\#\mathsf{P}/\mathrm{poly}$. We have access to f only using

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

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 $\#\mathsf{P}$ 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 $\#\mathsf{P}/\mathrm{poly}$. Valiant's criterion (Proposition 2.1) now implies that the polynomial f is indeed in VNP.

2.2 Factor closure over *all* fields

The two classical paradigms involved in factoring multivariate polynomials are *Hensel lifting* and *Newton iteration* (see, e.g. [vzG84, vzGG13]), 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 [vzGK85, Len99, Gre16, BSV20], polynomials with bounded-depth circuits [DSY10, Oli16] and bounded individual degree [Oli16], algebraic branching programs [KK08, Jan11, ST21] and even classes beyond VP such as VNP [CKS19b] and polynomials of exponential degree [DSS22], 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 [Bü00, KSS15, Oli16, CKS19a, DSS22] 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 [KI04, DSY10, AGS19, CKS19b, KST19, KS19, GKSS22], polynomial identity testing [SV10, KSS15], coding theory [Sud97, GS99], cryptography [CR88], proof complexity [FSTW21], convex optimization [Oli20] and more. See [FS15, Sax23] for an introduction and survey of polynomial factoring.

In a recent work, [CKS19b] showed that VNP is closed under factoring over fields of characteristic zero. 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 [CKS19a], 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 5.2) uses slight modifications of standard techniques developed over the years [Kal87, KSS15, CKS19b]. 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 the factors at the end. To finally show that the factor we obtain is in VNP, we use a one-shot analysis as in [CKS19b].

Over fields of characteristic zero, it can be shown that proving Case 2 is sufficient (see proof of [CKS19a, 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 5.1). Using this result, we can then handle all powers (Lemma 5.3).

All previous known techniques fail in the case where the exponent e is a prime power. In the special case when the number of variables is bounded, a recent work by Andrews [And20] shows how to extract p-th roots of polynomials without blowing up the circuit size. Inspired by the proof of Theorem 1, 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 5.4).

It was remarked in [MP08, Section 6] that the fact has been observed before in [Pé04], though we could not find a written reference [5]. 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 to convert the algebraic expression thus obtained to a Boolean #P/ poly circuit.

Since $f \in \mathsf{VNP}$, 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 transform, 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

In this section, we present the necessary background, formal definitions, and a few results that will be used throughout the paper. We begin by introducing counting complexity classes, followed by a discussion on structural results of algebraic complexity theory.

3.1 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 [Bü04, Section 4.3] and [KP11, Section 2.2]. For a more comprehensive discussion refer to [Pap94]. For a natural number r, $\langle r\rangle \in \{0,1\}^*$ denotes the binary encoding of r.

Definition 3.1 (#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 $\chi \in \mathsf{P}$ satisfying $\psi(\boldsymbol{x}) = \langle |S| \rangle$ where*

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

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 $\mathrm{poly}(|\boldsymbol{x}|)$.

It is easy to show that FP is contained in #P (refer [SK12, Lemma 8]).

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 3.2 (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 $n \in \mathbb{N}$, $|\alpha(n)| \le \mathrm{poly}(n)$. The following lemma shows that the complexity

[5] Perifel communicated to us a proof that over $\mathbb{Q}$, the coefficients of constant-free VNP families (see [Mal03]) are in GapP/ poly.

classes of our interest are closed under usual operations.

Lemma 3.3 (Closure Properties)**.** *For a positive integer r, consider a set of functions $\phi_1, \ldots, \phi_r$ in $\#\mathsf{P}/\mathsf{poly}$. Consider an input string $\boldsymbol{x} \in \{0,1\}^*$. Then the following closure properties can be shown:*

1. Addition and Multiplication: *Let $\phi_+(\boldsymbol{x}) := \sum_{i\in[r]} \phi_i(\boldsymbol{x})$ and $\phi_\times := \prod_{i\in[r]} \phi_i(\boldsymbol{x})$. Then, ϕ_+ and $\phi_\times$ are also in $\#\mathsf{P}/\mathsf{poly}$ for $r \le \mathrm{poly}(|\boldsymbol{x}|)$.*

2. Repeated Squaring: *For all $i \in [r]$, $\phi_i(\boldsymbol{x})^t$ is in $\#\mathsf{P}/\mathsf{poly}$ for $t \le 2^{\mathrm{poly}(|\boldsymbol{x}|)}$.*

3. Projection: *Let $\Phi_i(\boldsymbol{x}) := \sum_{\boldsymbol{b}\in\{0,1\}^\ell} \phi_i(\boldsymbol{x}, \boldsymbol{b})$, where $\ell \le \mathrm{poly}(|\boldsymbol{x}|)$. Then Φ_i is in $\#\mathsf{P}/\mathsf{poly}$.*

Proof. For every $\#\mathsf{P}/\mathsf{poly}$ function ϕ_i, let α_i be its advice function and χ_i be its associated language in P defining the counting set S_i, see Definition 3.1.

1. Addition and Multiplication: Define an advice function $\alpha(|\boldsymbol{x}|, \langle i\rangle) = \alpha_i(|\boldsymbol{x}|)$, and two sets as follows:

$$S_+ := \left\{(\boldsymbol{i}, \boldsymbol{y}) \in \{0,1\}^{\mathrm{poly}(|\boldsymbol{x}|)+\log r} \ : \ (\boldsymbol{x}, \alpha(|\boldsymbol{x}|, \boldsymbol{i}), \boldsymbol{y}) \in \chi_i\right\}, \text{ and}$$
$$S_\times := \left\{(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r) \in \{0,1\}^{r\,\mathrm{poly}(|\boldsymbol{x}|)} \ : \ \forall i \in [r], (\boldsymbol{x}, \alpha(|\boldsymbol{x}|, \langle i\rangle), \boldsymbol{y}_i) \in \chi_i\right\}.$$

   For input $\boldsymbol{x} \in \{0,1\}^*$, let $\bar{\alpha}(|\boldsymbol{x}|) = (\alpha(|\boldsymbol{x}|, \langle 1\rangle), \ldots, \alpha(|\boldsymbol{x}|, \langle r\rangle))$ be the advice function. Then, it is easy to verify that

$$\phi_+(\boldsymbol{x}) = \psi_+(\boldsymbol{x}, \bar{\alpha}(|\boldsymbol{x}|)) := \langle |S_+|\rangle, \text{ and}$$
$$\phi_\times(\boldsymbol{x}) = \psi_\times(\boldsymbol{x}, \bar{\alpha}(|\boldsymbol{x}|)) := \langle |S_\times|\rangle.$$

   Due to the upper bound on r, the length of the advice string $\bar{\alpha}$ is bounded by $\mathrm{poly}(|x|)$. Moreover, ψ_+ and $\psi_\times$ are in $\#\mathsf{P}$ by definition. Hence, ϕ_+ and $\phi_\times$ are in $\#\mathsf{P}/\mathsf{poly}$.

2. Repeated Squaring: Note that $\phi_i(\boldsymbol{x})^2$ is in $\#\mathsf{P}/\mathsf{poly}$ from the discussion on multiplication above. Then the claim follows by repeatedly multiplying $\#\mathsf{P}/\mathsf{poly}$ function, $\log r$ many times.

3. The proof is in the same line as *addition*, which was discussed earlier. Since the advice function depends solely on the length of the input $\boldsymbol{x}$, it will be same throughout the hypercube-sum. This essentially, lets us add exponentially many $\#\mathsf{P}/\mathsf{poly}$ function. Let $\Psi_i(\boldsymbol{x}, \alpha_i(|\boldsymbol{x}|)) = \langle |S_{\mathsf{P}}|\rangle$ where

$$S_{\mathsf{P}} := \left\{(\boldsymbol{b}, \boldsymbol{y}) \in \{0,1\}^{\mathrm{poly}(|\boldsymbol{x}|)+\ell} \ : \ (\boldsymbol{x}, \alpha_i(|\boldsymbol{x}|), \boldsymbol{b}, \boldsymbol{y}) \in \chi_i\right\}.$$

Given the advice string $\alpha_i(|\boldsymbol{x}|)$ as input, clearly Ψ_i is in $\#\mathsf{P}$. Observe that $\Phi_i(\boldsymbol{x}) = \Psi_i(\boldsymbol{x}, \alpha_i(|\boldsymbol{x}|))$, hence Φ_i belongs to $\#\mathsf{P}/\mathsf{poly}$.

$\square$

In the proof of Lemma 4.2, we claimed that string function in $\#\mathsf{P}/\mathsf{poly}$ are closed under concatenation. In the following claim we formalise it. For binary string $\boldsymbol{a}, \boldsymbol{b} \in \{0,1\}^*$, we use $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ to denote string concatenation.

Claim 3.4. *For a positive integer r, consider a set of functions $\phi_i : \{0,1\}^* \to \{0,1\}^\ell$ in $\#\mathsf{P}/\mathsf{poly}$. Define a map $\Phi : \{0,1\}^* \to \{0,1\}^{r\ell}$ such that $\Phi(\boldsymbol{x}) = \langle \phi_1(\boldsymbol{x}), \ldots, \phi_r(\boldsymbol{x}) \rangle$, for all $\boldsymbol{x} \in \{0,1\}^*$. Then, Φ is in $\#\mathsf{P}/\mathsf{poly}$ for $r, \ell \leq \mathrm{poly}(|\boldsymbol{x}|)$.*

Proof. We can express the concatenation using the following expression.

$$\Phi(\boldsymbol{x}) = \left\langle \sum_{i \in [r]} \phi_i(\boldsymbol{x}) \cdot 2^{(r-i)\cdot \ell} \right\rangle .$$

Computing $2^{(r-i)\cdot\ell}$ is vacuously in $\mathsf{FP} \subseteq \#\mathsf{P}$. Using Lemma 3.3 finishes the proof. $\square$

3.2 Structural Results

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

Homogenisation. For a degree-d polynomial f, we denote its degree-k homogeneous components by $\mathrm{Hom}_k(f)$. Similarly, we define $\mathrm{Hom}_{\leq k}(f)$ as $\sum_{i\in[k]} \mathrm{Hom}_k(f)$. The following well-known structural result proves that, given a blackbox access to a circuit that computes the polynomial f, we can construct a circuit that computes all its homogeneous components. Refer to [SY10, Theorem 2.2] for the proof.

Lemma 3.5 (Homogenisation)**.** *Consider an n-variate polynomial $f := \sum_{i\in[d]} c_i(\boldsymbol{y}) \cdot x^i \in \mathbb{F}[\boldsymbol{y}][x]$ computable by a circuit of size s over $\mathbb{F}$. Then $\mathrm{size}(c_i)$ is at most $\mathrm{poly}(s, n, d)$, for all $i \in [d]$. Moreover, $\mathrm{size}(\mathrm{Hom}_{\leq d}(f))$ is at most $\mathrm{poly}(s, n, d)$.*

A straightforward application of the homogenisation lemma is the elimination of division gates and computing derivatives. Refer to [SY10, Theorem 2.11] for the details.

Lemma 3.6 (Division Elimination)**.** *Consider an n-variate polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ computable by a circuit of size s over $\mathbb{F}$. Then $f \mod \langle \boldsymbol{x} \rangle^{d+1}$ can be computed by a circuit of size $\mathrm{poly}(s, d)$.*

Lemma 3.7 (Derivatives)**.** *Consider an n-variate polynomial $f \in \mathbb{F}[y_1, \ldots, y_n][x]$ computable by a circuit of size s over $\mathbb{F}$. Then for any k, $\partial_x^k f$ can be computed by a circuit of size $\mathrm{poly}(s, k)$.*

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 [Val82] proved that these polynomials can be equivalently computed by a hypercube-sum of small size formulas. Refer to [Bü00, Theorem 2.13] and [MP08, Theorem 2] for the proof. A direct consequence of the equivalence is the following structural lemma, which helps to prove the closure properties of VNP.

Lemma 3.8 (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, where both the number of auxiliary variables m and the* formula *size of h both bounded by* $\text{poly}(s, n, d)$*, satisfying the hypercube-sum expression:*

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

Closure properties of VNP In Section 5 we discussed various closure properties of polynomials in the class VNP. We invoke these closure properties to proof Theorem 2. The Composition property is particularly crucial in the proof of Lemma 5.2. In a subsequent work, Valiant [Val82] 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 3.9 (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* $\text{poly}(n, m)$*. 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*.*

Proof. The closure properties can be proved using Definition 1.1 of VNP. For all $i \in [t]$, let $(m_i, \text{size}(h_i))$ be the size parameters for f_i in VNP over $\mathbb{F}$, where both the parameters and $\deg(f_i)$ are bounded by $\text{poly}(n, m)$. Then the properties can be proved as follows.

Addition and Multiplication: Observe that

$$\begin{aligned} f_+ = \sum_{i\in[t]} f_i &= \sum_{i\in[t]} \left(\sum_{\boldsymbol{a}_i\in\{0,1\}^{m_i}} h_i(\boldsymbol{x}, \boldsymbol{a}_i) \right) \\ &= \sum_{(\boldsymbol{a}_1,\ldots,\boldsymbol{a}_t)\in\{0,1\}^{\ell_+}} h_+(\boldsymbol{x}, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_t), \end{aligned}$$

where $\ell_+ := \sum_{i\in[t]} m_i$ and $h_+ := \sum_{i\in[t]} h_i(\boldsymbol{x}, \boldsymbol{a}_i)$. Since both t and m_i, are bounded by $\mathrm{poly}(n, m)$, the length of the witness ℓ_+ is atmost $\mathrm{poly}(n, m)$. Moreover, $\mathrm{size}(h_+) = 3 + t + \sum_{i\in[t]} \mathrm{size}(h_i) \leq \mathrm{poly}(n, m)$. Similarly for multiplication we have

$$\begin{aligned} f_\times &= \prod_{i\in[t]} f_i = \prod_{i\in[t]} \left(\sum_{\boldsymbol{a}_i\in\{0,1\}^{m_i}} h_i(\boldsymbol{x}, \boldsymbol{a}_i) \right) \\ &= \sum_{(\boldsymbol{a}_1,\ldots,\boldsymbol{a}_t)\in\{0,1\}^{\ell_\times}} h_\times(\boldsymbol{x}, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_t). \end{aligned}$$

A similar analysis reveals that VNP size parameters $(\ell_\times, \mathrm{size}(h_\times))$ of $f_\times$ are bounded by $\mathrm{poly}(n, m)$.

Coefficient Extraction: The proof runs the same as the proof of Lemma 4.1, with both s and D at most $\mathrm{poly}(n, m)$. Note that standard interpolation using random evaluation points would suffice only for fields of large characteristics.

Composition: We will follow the proof sketch of [CKS19b, Claim 8.4]. Suppose that g is hypercube sum of verifier polynomials v. It is enough to prove the statement for $v \in \mathsf{VP}$. Invoke Lemma 3.8 on the circuit C for the verifier polynomial v to obtain a polynomial h and $\ell \leq \mathrm{poly}(t, d)$ satisfying

$$C = \sum_{\boldsymbol{a}\in\{0,1\}^\ell} h(z_1, \ldots, z_t, a_1, \ldots, a_\ell).$$

Let T be the formula computing h of size $\mathrm{poly}(t, d)$. Composing with the VNP polynomials gives

$$C(f_1, \ldots, f_t) = \sum_{\boldsymbol{a}\in\{0,1\}^\ell} h(f_1, \ldots, f_t, a_1, \ldots, a_\ell).$$

We claim that feeding the verifier circuits h_i of the VNP polynomials f_i, into the formula T gives the required hypercube-sum representation.

$$C(f_1, \ldots, f_t) = \sum_{\boldsymbol{a},\boldsymbol{a}_i\in\{0,1\}^{\ell'}} T(h_1(\boldsymbol{x}, \boldsymbol{a}_i), \ldots, h_t(\boldsymbol{x}, \boldsymbol{a}_t), \boldsymbol{a}),$$

where $\ell' = \ell + \sum_i m_i \leq \mathrm{poly}(t, d, n, m)$. Moreover, the size of the circuit computing T composed with $h_1, \ldots, h_t$ is at most $O(\mathrm{size}(T) + \sum_i \mathrm{size}(h_i)) \leq \mathrm{poly}(t, d, n, m)$. The correctness of the expression above, follows from an easy induction on the depth of the formula T. Along the layers, from bottom to the top, we repeatedly invoke the additive and multiplicative closure properties which were discussed earlier. Since T is a formula, the verifier circuits for each of the h_i's receive their unique copy of the witnesses and this is preserved throughout the computation. The last part is crucial for the correctness because simply plugging in the h_i's to the *circuit* C could result in the same witnesses being reused and it may not be the intended computation [6]. □

[6] Consider a pedagogical example, $C(z) = z^2$ from [CKS19b, Footnote 9].

4 Presentable is explicit

In this section we will give the prove of Theorem 1, that the polynomials in $\overline{\mathsf{VNP}}_\varepsilon$ are explicit over finite fields. 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 $s := \mathrm{poly}(r, \log q)$ and let $g = \sum_{\boldsymbol{e}} c_{\boldsymbol{e}} \boldsymbol{y}^{\boldsymbol{e}}$ be an r-variate polynomial over $\mathbb{F}_q$ of degree $D := \exp(s)$ 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 $\mathrm{poly}(s)$.

We will prove the above lemma in Section 4.1. In the subsequent lemma we show that the resulting hypercube sum above can be converted into a boolean function in $\#\mathsf{P}/\mathsf{poly}$. 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). *For any exponent vector $\boldsymbol{e} \in \{0, \ldots, D\}^r$, 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:*

$$c_{\boldsymbol{e}} = \sum_{\boldsymbol{b} \in \{0,1\}^\ell} t_{\boldsymbol{e}}(b_1, \ldots, b_\ell) \;, \tag{4.3}$$

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 $\#\mathsf{P}/\mathsf{poly}$ that computes $\phi_g(\langle \boldsymbol{e} \rangle) = \langle c_{\boldsymbol{e}} \rangle$.

We will defer the proof of the lemma until Section 4.2. Meanwhile, we will use the technical lemmas to give the complete proof of our first main result.

Theorem 1 (Presentable is Explicit). *Over* any *finite field,* $\overline{\mathsf{VNP}}_\varepsilon = \mathsf{VNP}$.

Proof. Consider a polynomial (family) $f = f_n \in \mathbb{F}_q[x_1, \ldots, x_n]$ in $\overline{\mathsf{VNP}}_\varepsilon$ 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 $\overline{\mathsf{VNP}}_\varepsilon$ 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 $\#\mathsf{P}/\mathsf{poly}$ 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

$$f = \sum_{\boldsymbol{e} \in \{0,\ldots,d\}^n} c_{\boldsymbol{e}} \cdot \boldsymbol{x}^{\boldsymbol{e}}, \tag{4.4}$$

then $\langle c_{\boldsymbol{e}} \rangle = \phi_g(M, e_1, \ldots, e_n)$. From the definition of $\overline{\mathsf{VNP}}_\varepsilon$, 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. $\square$

4.1 Exponential interpolation technique

In this section we will give the proof of Lemma 4.1. We will show that the coefficients of the polynomial g from the lemma statement can be expressed as a hypercube sum of evaluation of small size circuits. Recall the size parameter $s = \mathrm{poly}(r + m, \log q)$ and $q =: p^a$ for prime p. We will induct on the number of variables r.

Consider a positive integer k such that $2^s = D < k < \Theta(D)$, and a primitive root of unity ω of order k. We know that $\omega \in \mathbb{F}_q$ if and only if k divides $q - 1$ (refer [vzGG13, Lemma 8.8]). Moreover, if $\mathbb{F}_q$ does not contain the particular primitive root of unity, we can obtain them in the multiplicative group of its finite field extension $\mathbb{F}_{q'}$, where $k < q' := p^{a'} = \Theta(D)$. Interested readers are encouraged to read more details in standard literature on Finite Fields, for instance refer to [vzGG13, Chapter 8] and [Sho09, Exercise 17.24]. For the rest of the section we will assume for simplicity that $\omega \in \mathbb{F}_q$; as an identical proof works over the extension $\mathbb{F}_{q'}$. Note that $1/k \in \mathbb{F}_q$, as $k|(q-1)$ implies that $p \nmid k$.

Base case. Suppose g is a univariate polynomial in $y = y_1$ and consider an exponent $e \leq k$. To extract the coefficient c_e in g, we will interpolate by evaluating g on a set of k distinct points $\{\omega^0, \omega^1, \ldots, \omega^{k-1}\}$, constituting all the powers of this primitive root of unity. These evaluations of g form a linear system using Vandermonde matrix $V_\omega := \left(\omega^{ij}\right)_{0 \leq i,j < k}$ as follows:

$$\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{k-1} & \omega^{2(k-1)} & \cdots & \omega^{(k-1)^2} \end{pmatrix}_{k \times k} \begin{pmatrix} \vdots \\ c_e \\ \vdots \end{pmatrix}_{k \times 1} = \begin{pmatrix} \vdots \\ g\left(\omega^e\right) \\ \vdots \end{pmatrix}_{k \times 1} .$$

Vandermonde matrices are invertible if and only if its entries are all distinct. It is clear that ω^{-1} is also a primitive root of unity; moreover, the inverse matrix $(V_\omega)^{-1} = (1/k) \cdot V_{(\omega^{-1})}$ (refer [vzGG13, Theorem 8.13]). Therefore, we can express the required coefficient with the following equation:

$$
\begin{aligned}
c_e &= \sum_{j=0}^{k-1} \frac{\omega^{-ej}}{k} \cdot g\left(\omega^j\right) \\
&= \sum_{\boldsymbol{a}\in\{0,1\}^m} \left(\sum_{j=0}^{k-1} \frac{\omega^{-ej}}{k} \cdot h\left(\omega^j, \boldsymbol{a}\right) \right) .
\end{aligned} \tag{4.5}
$$

A circuit that computes the inner sum in Equation 4.5 trivially, would be exponentially large in s because $k = \Theta(D)$. However, we can write this as a hypercube-sum by carefully encoding the powers of ω in a single polynomial using binary representation of the exponent. This encoding will design a verifier circuit, with a relatively small increase in the witness size. Let $\mathrm{wt}(k) := \lceil \log_2 k \rceil$ and use it to define a polynomial $\bar{h} \in \mathbb{F}_q[z, z_1, \ldots, z_{\mathrm{wt}(k)}]$ as follows:

$$
\bar{h} \;:=\; \prod_{i=1}^{\mathrm{wt}(k)} \left(z_i \cdot z^{2^{i-1}} + (1 - z_i) \cdot 1 \right) . \tag{4.6}
$$

Let $\boldsymbol{j} := (j_1, \ldots, j_{\mathrm{wt}(k)})$ be the binary representation of j, then it is easy to verify that $\bar{h}(\omega, \boldsymbol{j}) = \omega^j$. Together with $\bar{h}$, Equation 4.5 can be re-written as follows:

$$
\begin{aligned}
c_e \;&=\; \sum_{\boldsymbol{a}\in\{0,1\}^m} \; \sum_{\boldsymbol{j}\in\{0,1\}^{\mathrm{wt}(k)}} \frac{1}{k} \cdot \bar{h}(\bar{h}\left(\omega^{-1}, \langle e \rangle\right), \boldsymbol{j}) \cdot h\left(\bar{h}(\omega, \boldsymbol{j}), \boldsymbol{a}\right) \\
&=: \sum_{\boldsymbol{a},\boldsymbol{j}\in\{0,1\}^{\ell}} t_e(\boldsymbol{a}, \boldsymbol{j}) \;,
\end{aligned}
$$

where $\ell := m + \mathrm{wt}(k) \le O(s)$. Observe that $\mathrm{size}\left(\bar{h}\right) \le O(\mathrm{wt}(k)) \le O(s)$, moreover, composition and multiplication have additive blow-up on size of the circuit. Since, $\mathrm{size}(h)$ was bounded by s, overall gluing the circuits together shows that $\mathrm{size}(t_e) \le O(s)$.

Induction step. Let us assume that the lemma holds for all such $r-1$ variate polynomials. Now, suppose g is a r-variate polynomial in $\mathbb{F}_q[y_2, \ldots, y_r][y_1]$ such that

$$
g = \sum_{i \le D} g_i(y_2, \ldots, y_n) \cdot y_1^i,
$$

where g_i is $(r-1)$-variate polynomial of degree at most D. With respect to the fixed exponent vector $\boldsymbol{e} = (e_1, e_2, \ldots, e_r) \in \mathbb{N}^r$, define $\boldsymbol{e}^- := (e_2, \ldots, e_r) \in \mathbb{N}^{r-1}$. From the equation above, observe that computing the coefficient $c_{\boldsymbol{e}}$ of $y_1^{e_1} y_2^{e_2} \cdots y_r^{e_r}$ in g is equivalent to computing the coefficient $c_{\boldsymbol{e}^-}$ of $y_2^{e_2} \cdots y_r^{e_r}$ in g_{e_1}. To invoke the induction hypothesis on g_{e_1}, we first need to show that, like g, it can be explicitly expressed as a hypercube-sum of a small sized circuit.

Once again interpolate on g to obtain the coefficient of $y_1^{e_1}$. Similar to the *base case*, begin by considering the evaluations of g on the set of powers $\left\{\omega^0, \omega^1, \ldots, \omega^{k-1}\right\}$. The equivalent linear system obtained using the Vandermonde matrix V_ω is as follows:

$$\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{k-1} & \omega^{2(k-1)} & \cdots & \omega^{(k-1)^2} \end{pmatrix}_{k\times k} \begin{pmatrix} \vdots \\ g_{e_1} \\ \vdots \end{pmatrix}_{k\times 1} = \begin{pmatrix} \vdots \\ g\left(\omega^{e_1}, \boldsymbol{y}\right) \\ \vdots \end{pmatrix}_{k\times 1} .$$

As argued earlier, the matrix is invertible; more importantly, its elements are easily obtained from $(V_\omega)^{-1} = (1/k) \cdot V_{(\omega^{-1})}$. This results in the following expression for the $(r-1)$-variate coefficient polynomial:

$$\begin{aligned} g_{e_1} &= \sum_{j=0}^{k-1} \frac{\omega^{-e_1 j}}{k} \cdot g\left(\omega^j, y_2, \ldots, y_r\right) \\ &= \sum_{\boldsymbol{a}\in\{0,1\}^m} \left(\sum_{j=0}^{k-1} \frac{\omega^{-e_1 j}}{k} \cdot h\left(\omega^j, y_2, \ldots, y_r, \boldsymbol{a}\right) \right). \end{aligned} \tag{4.7}$$

To show that the inner summation has small size circuit, we encode the powers of root of unity using the polynomial $\bar{h}$ defined in Equation 4.6. All together, it gives the following compact expression:

$$\begin{aligned} g_{e_1} &= \sum_{\boldsymbol{a}\in\{0,1\}^m} \sum_{\boldsymbol{j}\in\{0,1\}^{\mathrm{wt}(k)}} \frac{1}{k} \cdot \bar{h}(\bar{h}(\omega^{-1}, \langle e_1 \rangle), \boldsymbol{j}) \cdot h\left(\bar{h}(\omega, \boldsymbol{j}), y_2, \ldots, y_r, \boldsymbol{a}\right) \\ &=: \sum_{\boldsymbol{a},\boldsymbol{j}\in\{0,1\}^\ell} h_{e_1}(y_2, \ldots, y_r, \boldsymbol{a}, \boldsymbol{j}) , \end{aligned} \tag{4.8}$$

where $\ell := m + \mathrm{wt}(k) \le O(s)$. Further, $\mathrm{size}(h_{e_1}) \le s + 2 \times \mathrm{size}(\bar{h}) \le O(s)$.

Size analysis. We analyse the size of the verifier-circuit of c_e by unfolding the induction layers. Since g_{e_1} is now a $(r-1)$-variate polynomial, using induction hypothesis we get that there is a polynomial $t_{\boldsymbol{e}}$, that computes the relevant coefficient $c_{\boldsymbol{e}^-}$ as follows:

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

whence we define $s(r) := \mathrm{size}(t_{\boldsymbol{e}})$. Building on the insights from Equation 4.8, observe that in each iteration of the interpolation, the verifier is only evaluated and multiplied by the polynomial $\bar{h}$. This stems from the nature of interpolation, which extracts the coefficients as a linear combination of polynomial evaluations. So, we get a simple recurrence: $s(r) \le s(r-1) + 2 \cdot \mathrm{size}(\bar{h})$, which implies that the final verifier-circuit size $s(r) \le O(rs)$. Analogously, the witness length increases by $\mathrm{wt}(k)$ in each iteration, hence $\ell(r) \le m + r \cdot \mathrm{wt}(k) \le O(rs)$. That concludes the proof of Lemma 4.1. □

4.2 Transfer algebraic complexity to boolean

In this section, we will show that the hypercube-sum of the evaluations of a small-size circuit can be transformed into a $\#\mathsf{P}/\mathsf{poly}$ function, which will prove Lemma 4.2. As described earlier, the proof goes via *booleanisation* of the algebraic circuit. Recall that $q = p^a$, and for a field element $b \in \mathbb{F}_q$, $\langle b \rangle \in \{0,1\}^s$ denotes the binary encoding of b. For a point $\boldsymbol{b} \in \mathbb{F}_q^{\ell}$, denote $\langle \boldsymbol{b} \rangle := (\langle b_1 \rangle, \ldots, \langle b_\ell \rangle) \in \{0,1\}^{\ell s}$.

Claim 4.9 (Booleanisation)**.** *Consider a polynomial $t \in \mathbb{F}_q[y_1, \ldots, y_\ell]$ such that* $\mathrm{size}(t) \leq s$*. There exists an equivalent (multi-output) boolean circuit T of* bitsize $\leq s \cdot \mathrm{poly}(\log q)$*, such that for all inputs $\boldsymbol{b} \in \mathbb{F}_q^{\ell}$ we have $T(\langle \boldsymbol{b} \rangle) = \langle t(\boldsymbol{b}) \rangle$.*

Proof. Let C be an algebraic circuit of size at most s which computes the polynomial t. Without loss of generality. we assume that the circuit has fan-in two. The idea is to build a Boolean circuit from the Algebraic circuit by replacing each of its field operation gates with equivalent Boolean gadgets. Following is a formal proof of it using induction on the depth of C.

In the base case, we have variables and constants at the input level. To construct the equivalent Boolean circuit T, split every input variable y_i into $\log q$ many gates which takes $\langle b_i \rangle$ as input. Similarly, every constant β in $\mathbb{F}_q$ can be split into $\log q$ many gates based on $\langle \beta \rangle$. Therefore $\mathrm{bitsize}(T) \leq O(s \cdot \log q)$.

Let C_1, C_2 be sub-circuits of C, connected to an internal node C_{12}. From the induction hypothesis, there are equivalent Boolean circuits T_1, T_2 of bitsize at most $s \cdot \mathrm{poly}(\log q)$ such that for all inputs $\boldsymbol{b} \in \mathbb{F}_q^{\ell}$ we get $T_i(\langle \boldsymbol{b} \rangle) = \langle C_i(\boldsymbol{b}) \rangle$, for $i \in [2]$. Arithmetic operations in a finite field, for instance, addition and multiplication, can be efficiently simulated by Boolean circuits (that have input and output as binary strings). In particular, there are $\mathrm{poly}(\log q)$ size Boolean circuits T_+ and $T_\times$ such that for all $b_1, b_2 \in \mathbb{F}_q$, $\langle b_1 + b_2 \rangle = T_+(\langle b_1 \rangle, \langle b_2 \rangle)$ and $\langle b_1 \times b_2 \rangle = T_\times(\langle b_1 \rangle, \langle b_2 \rangle)$[7]. For a detailed discussion on computational complexity of finite field arithmetic refer [GS11, Section 2] and [AB09, Section A.4].

Based on the gate C_{12}, use either T_+ or $T_\times$ with T_1 and T_2 as inputs to obtain the circuit T_{12} such that for all inputs $\boldsymbol{b} \in \mathbb{F}_q^{\ell}$ we have $T_{12}(\langle \boldsymbol{b} \rangle) = \langle C_{12}(\boldsymbol{b}) \rangle$. Notice that $\mathrm{bitsize}(T_{12}) = \mathrm{bitsize}(T_1) + \mathrm{bitsize}(T_2) + \max(\mathrm{bitsize}(T_+), \mathrm{bitsize}(T_\times))$. Proceeding this way in a level-by-level fashion, we obtain the complete Boolean circuit T which computes $\langle t(\boldsymbol{b}) \rangle$. Finally, for the bitsize claim we observe that every gate is replaced by either T_+ or $T_\times$ and thus $\mathrm{bitsize}(T) \leq s \cdot \max(\mathrm{bitsize}(T_+), \mathrm{bitsize}(T_\times)) \leq s \cdot \mathrm{poly}(\log q)$. □

We will use the above claim to convert the algebraic circuit in the hypercube sum of the coefficient into an efficiently computable Boolean function. Recall the hypercube-sum expression

[7]The Boolean encoding and the output of Boolean circuit are compared coordinate-wise.

for coefficients from Lemma 4.2:

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

where ℓ and $\mathrm{size}(t_{\boldsymbol{e}})$ are at most $\mathrm{poly}(s)$. Since $c_{\boldsymbol{e}}$ and $t_{\boldsymbol{e}}(\boldsymbol{b})$ are elements of $\mathbb{F}_q$, their binary representation is an encoding of tuple of $\mathbb{F}_p$ elements. Refer the remark following Proposition 2.1.

Proof of Lemma 4.2. Consider the $\mathbb{F}_p$-basis representation of $\mathbb{F}_q$ element $t_{\boldsymbol{e}}(\boldsymbol{b}) = \sum_{i<a} t_{\boldsymbol{e},i}\alpha^i$, where $t_{\boldsymbol{e},i} \in \mathbb{F}_p$. Apply Claim 4.9 to the algebraic circuit that computes $t_{\boldsymbol{e}}$ to obtain a multi-output equivalent Boolean circuit $T_{\boldsymbol{e}}$ satisfying $\langle t_{\boldsymbol{e}}(\boldsymbol{b})\rangle = T_{\boldsymbol{e}}(\langle \boldsymbol{b}\rangle)$, for all $\boldsymbol{b} \in \{0,1\}^{\ell}$. The Boolean circuit $T_{\boldsymbol{e}}$ computes the encoding of $\mathbb{F}_q$ element as a tuple $(\langle t_{\boldsymbol{e},0}\rangle, \ldots, \langle t_{\boldsymbol{e},a-1}\rangle)$. Let $T_{\boldsymbol{e},i}$ denote a sub-circuit of $T_{\boldsymbol{e}}$ computing the string $\langle t_{\boldsymbol{e},i}\rangle$.

We claim that $T_{\boldsymbol{e},i}$ is in the complexity class $\mathsf{FP/poly}$ (refer Section 3.1 for definitions). Define a Turing Machine M that takes $\langle T_{\boldsymbol{e}}\rangle$ as advice, and evaluates $T_{\boldsymbol{e}}$ at the input $\langle \boldsymbol{b}\rangle$ in time $\mathrm{poly}(s)$, for any $\boldsymbol{b} \in \{0,1\}^{\ell}$. The size of the advice string $\langle T_{\boldsymbol{e}}\rangle$ is independent of the input and depends only on the input length $\ell \le \mathrm{poly}(s)$. Finally, the Turing machine outputs the i-th block of the evaluation. Clearly, the function computed by M is in FP, and hence $T_{\boldsymbol{e},i}$ is in $\mathsf{FP/poly}$.

Let the $\mathbb{F}_p$-basis representation of the coefficient be $c_{\boldsymbol{e}} = \sum_{i<a} c_{\boldsymbol{e},i}\alpha^i$, where $c_{\boldsymbol{e},i} \in \mathbb{F}_p$. To design the coefficient function ϕ_g that computes the encoding of $c_{\boldsymbol{e}}$, it suffices to prove that there is a function $\phi_{g,i}(\langle \boldsymbol{e}\rangle)$ in $\#\mathsf{P/poly}$ that computes $\langle c_{\boldsymbol{e},i}\rangle$, for all $i < a$ (see Claim 3.4 and remark of Proposition 2.1). From Equation 4.3, we see that $c_{\boldsymbol{e},i} = \sum_{\boldsymbol{b}\in\{0,1\}^{\ell}} t_{\boldsymbol{e},i}$, where the sum is over $\mathbb{F}_p$. Therefore, we can express $\langle c_{\boldsymbol{e},i}\rangle$ as a hypercube sum of $\langle t_{\boldsymbol{e},i}\rangle$ reduced modulo p, and thus, also as a hypercube-sum of $T_{\boldsymbol{e},i}(\langle \boldsymbol{b}\rangle)$, modulo prime p. Using Lemma 3.3(3) obtain a $\#\mathsf{P/poly}$ function that computes this hypercube-sum. It gives our desired coefficient function $\phi_{g,i}(\langle \boldsymbol{e}\rangle) \in \#_p\mathsf{P/poly}$ (refer to remark of Proposition 2.1). $\square$

4.3 Application: Deborder Factors

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

Definition 4.10 (Presentable $\overline{\mathsf{VP}}$)**.** *The presentable border class $\overline{\mathsf{VP}}_\varepsilon$ 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 emphasize that the degree of ε-polynomials in g is unrestricted. Further, it is apparent from the definitions that $\mathsf{VP} \subseteq \overline{\mathsf{VP}}_\varepsilon \subseteq \overline{\mathsf{VNP}}_\varepsilon$. We note that a form of *semicontinuity* (cf. [Bü04, Lemma 2.3]) also holds with our definition.

Claim 4.11. *Suppose that $f(\boldsymbol{x})$ can be approximated by a polynomial $g(\boldsymbol{x},\delta) \in \mathbb{F}[\delta][x_1,\ldots,x_n]$ that is in $\overline{\mathsf{VP}}_\varepsilon$. Then, $f \in \overline{\mathsf{VP}}_\varepsilon$.*

Proof. Since $g(\boldsymbol{x},\delta) \in \overline{\mathsf{VP}}_\varepsilon$, there is an approximating polynomial h such that

$$h(\boldsymbol{x},\delta,\varepsilon) = \varepsilon^M g(\boldsymbol{x},\delta) + \varepsilon^{M+1} Q(\boldsymbol{x},\delta,\varepsilon) \tag{4.12}$$

where $\text{size}_{\mathbb{F}[\delta]}(h) = \text{poly}(n)$, and $\deg_{\boldsymbol{x},\delta}(h) = \text{poly}(n)$. Since h has at most $\text{poly}(n)$ edges labeled with $\text{poly}(n)$-degree univariates in δ, this also means that $\text{size}_{\mathbb{F}}(h) = \text{poly}(n)$. As f can be approximated by g,

$$g(\boldsymbol{x},\delta) = \delta^{M'} f(\boldsymbol{x}) + \delta^{M'+1} Q'(\boldsymbol{x},\delta). \tag{4.13}$$

Now, substitute for g in Equation 4.12 using Equation 4.13, and set $\varepsilon = \delta^{M'+1}$. We get

$$h(\boldsymbol{x},\delta,\delta^{M'+1}) = \delta^{M(M'+1)+M'} f(\boldsymbol{x}) + \delta^{M(M'+1)+M'+1} Q'(\boldsymbol{x},\delta) + \delta^{M(M'+1)+M'+1} Q(\boldsymbol{x},\delta,\delta^N).$$

Note that setting $\varepsilon = \delta^{M'+1}$ still retains $\text{size}_{\mathbb{F}}(h) = \text{poly}(n)$ since $M' = \text{poly}(n)$. Combined with the fact that $\deg_{\boldsymbol{x}}(h) = \text{poly}(n)$, this shows $f(\boldsymbol{x}) \in \overline{\mathsf{VP}}_\varepsilon$.

□

As a direct application of Claim 4.11, we obtain

Corollary 4.14. $\overline{\left(\overline{\mathsf{VP}}_\varepsilon\right)}_\varepsilon = \overline{\mathsf{VP}}_\varepsilon$, *and* $\overline{\left(\overline{\mathsf{VNP}}_\varepsilon\right)}_\varepsilon = \overline{\mathsf{VNP}}_\varepsilon$.

A factor is called *separable* when it is irreducible and has multiplicity coprime to the characteristic of the field. Bürgisser proved in [Bü04, Theorem 1.3], that the class $\overline{\mathsf{VP}}_\varepsilon$ 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 4.15. *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* $\text{size}(f)$ *and* $\deg(u)$ *is at most $s := \text{poly}(n, \log q)$. Then we have u in $\overline{\mathsf{VP}}_\varepsilon$.*

Remark. We make a few observations.

1. In case $f = u^e$, Kaltofen [Kal87] showed that u is VP.

2. Bürgisser [Bü04] proved that u (in the lemma above) is in $\overline{\mathsf{VP}}$. Moreover, he remarked that, in his proof, the required polynomials in $\mathbb{F}[\varepsilon]$ do have small circuit-complexity (refer the remark following [Bü04, Definition 2.1]). For the sake of completeness, we will sketch the proof for $u \in \overline{\mathsf{VP}}_\varepsilon$ in the appendix.

As an application of the debordering result over finite fields in Theorem 1, we prove that the low-degree separable factors of small size circuits are explicit.

Corollary 1.3 (Formally restated). *Let $q := p^a$ and e be a positive integer coprime to p. Consider a polynomial (family) $f \in \mathbb{F}_q[x_1,\ldots,x_n]$ and its irreducible factor u satisfying $f = u^e v$,*

u coprime to v, such that $\text{size}(f)$ *and* $\deg(u)$ *is* $\text{poly}(n, \log q)$*. Then, the polynomial (family) u is in* VNP.

Proof. We learn from Lemma 4.15 that the polynomial family $u \in \overline{\mathsf{VP}}_\varepsilon$. Moreover, $\overline{\mathsf{VP}}_\varepsilon$ is contained in $\overline{\mathsf{VNP}}_\varepsilon$ by definition. As over $\mathbb{F}_q$, Theorem 1 proves $\overline{\mathsf{VNP}}_\varepsilon = \mathsf{VNP}$, hence $u \in \mathsf{VNP}$. □

In the rest of the section we will sketch the proof of Lemma 4.15. Consider a polynomial $f \in \mathbb{F}_q[x_1, \ldots, x_n]$ of degree d_f from the lemma statement. For all $i \in [n]$, randomly pick field elements $\alpha_i, \beta_i \in_r \mathbb{F}_q$ and define a map $\tau : x_i \mapsto x_i + \alpha_i y + \beta_i$, where y is a new variable. Under such a random invertible transformation, the polynomial completely splits over power series ring, see [DSS22, Theorem 17]. In particular, there exists $k \in \mathbb{F}_q^*, \gamma_i > 0$ and $h_i \in \mathbb{K}[[x_1, \ldots, x_n]]$ satisfying

$$\tau(f) = k \cdot \prod_{i \in [d_f]} (y - h_i)^{\gamma_i},$$

where $\mathbb{K}$ is a field extension of $\mathbb{F}_q$ of degree at most d_f. Refer [DDS21, Section 3 and 6.2] for details. Further, $\mu_i := h_i\left(\overline{0}\right)$ are all distinct nonzero field elements. We assume $\mathbb{K} = \mathbb{F}_q$ without loss of generality (refer [Bü00, Proposition 4.1]). An immediate corollary of such a power series split is the following lemma (refer [DSS22, Corollary 18])

Lemma 4.16. *Let u be a factor of f of degree d_u, and $\tau(f)$ splits as before. Since $\tau(u)$ divides $\tau(f)$, we deduce that*

$$\tau(u) = k' \cdot \prod_{i \in [d_u]} (y - h_i)^{c_i},$$

where $0 \le c_i \le \gamma_i$, $k' \in \mathbb{F}_q^*$, and $h_i \in \mathbb{F}_q[[x_1, \ldots, x_n]]$.

Recall the definition of $\text{Hom}_{\le d_u}(h_i)$ from Section 3. Observe that

$$\tau(u) \equiv k' \cdot \prod_{i \in [d_u]} (y - \text{Hom}_{\le d_u}(h_i))^{c_i} \bmod \langle x_1, \ldots, x_n \rangle^{d_u+1}.$$

Later we will show that due to the expression above it would suffice to give a complexity bound of the power series roots of $\tau(f)$ to uniquely recover the factor $\tau(u)$. The following proposition proves that all the power series roots can be easily approximated, see [Bü04, Proposition 3.4].

Proposition 4.17. *For all $i \in [d_f]$, there is an approximating polynomial $g_i \in \mathbb{F}_q[\varepsilon][x_1, \ldots, x_n]$ satisfying $g_i = \varepsilon^M \text{Hom}_{\le d_u}(h_i) + \varepsilon^{M+1} Q_i(\boldsymbol{x}, \varepsilon)$, for some error $Q_i \in \mathbb{F}[\varepsilon][x_1, \ldots, x_n]$ and order $M \in \mathbb{N}$. Moreover* $\deg_{\boldsymbol{x}}(g_i) \le d_u$ *and* $\text{size}_{\mathbb{F}}(g_i) \le \text{poly}(d_u, \text{size}(f))$*.*

Proof Sketch. Let $\tilde{f} := \tau(f) \in \mathbb{F}_q[\boldsymbol{x}, y]$ and $\mu_i := h_i(\overline{0})$. Define a perturbed polynomial $F := \tilde{f}(\boldsymbol{x}, y + \mu_i + \varepsilon) - \tilde{f}(0, \mu_i + \varepsilon)$ over the ring $\mathbb{F}_q[\varepsilon]$. Since e is coprime to p, with appropriate coordinate shift it can be ensured that $F|_{\varepsilon=0} = \tilde{f}$, $F(\overline{0}, 0) = 0$, but $\partial_y F(\overline{0}, 0) \neq 0$, see [Bü04, Equation 5]. Then the power series root H_j of the perturbed polynomial F can be obtained by classical Newton Iteration (refer [DSS22, Lemma 15]) as follows:

$$y_0 = 0, \quad y_{t+1} \coloneqq y_t - \frac{F(\boldsymbol{x}, y_t)}{\partial_y F(\boldsymbol{x}, y_t)} \tag{4.18}$$

where $y_t \equiv H_j \bmod \langle \boldsymbol{x} \rangle^{2^t}$. The quadratic convergence of degree in Newton iteration implies that it suffices to assume $t \le \log d_u + 1$. An easy induction on t proves that y_t, and therefore H_j, is well defined over $\mathbb{F}_q[[\varepsilon]]$. Hence, $H_j|_{\varepsilon=0} = h_i$, moreover $\mathrm{Hom}_{\le d_u}(H_j)|_{\varepsilon=0} = \mathrm{Hom}_{\le d_u}(h_i)$.

Let R be the subring of $\mathbb{F}(\varepsilon)$ consisting of rational functions defined at $\varepsilon = 0$. The preceding discussion then proves that an approximating polynomial $\tilde{g}_i \in \mathsf{R}[\boldsymbol{x}]$ computes $A_t/B_t \equiv y_t \bmod \langle \boldsymbol{x} \rangle^{2^t}$ and satisfies the following:

$$\tilde{g}_i = \mathrm{Hom}_{\le d_u}(h_i) + \varepsilon\, \tilde{Q}_i(\varepsilon, \boldsymbol{x}, y),$$

for some error $\tilde{Q}_i \in \mathsf{R}[x_1, \ldots, x_n]$. Bürgisser proved in [Bü04, Lemma 5.6] that equivalently there exists an approximating polynomial $g_i \in \mathbb{F}[\varepsilon][x_1, \ldots, x_n]$, order $M \in \mathbb{N}$, and error Q_i, such that $g_i = \varepsilon^M \mathrm{Hom}_{\le d_u}(h_i) + \varepsilon^{M+1} Q_i(\varepsilon, \boldsymbol{x}, y)$. Moreover, $g_i = \varepsilon^M \tilde{g}_i$. Therefore, the proposition follows easily by proving $\mathrm{size}_{\mathbb{F}}(\tilde{g}_i) \le \mathrm{poly}(d_u, \mathrm{size}(\tilde{f}))$, and $M \le 2^{\mathrm{poly}(d_u)}$. In case, $\deg_{\boldsymbol{x}}(g_i)$ is greater than d_u, homogenise and truncate the higher degree terms ([Bü04, Proposition 3.1]).

Size analysis. The circuit computing A_t/B_t is build iteratively using division gates following Equation 4.18. Treating ε as a variable, observe that $\mathrm{size}_{\mathbb{F}}(F) \le s_0 \coloneqq \mathrm{size}(\tilde{f}) + 2$. Homogenise the circuit computing F using Lemma 3.5 with respect to y to obtain $\mathrm{Hom}_{\le d_u}(F)$ of size $\mathrm{poly}(d_u, s_0)$. Use this homogenised circuit to obatin the circuit computing $\partial_y F$ of size $\mathrm{poly}(d_u, s_0)$. Using division and subtraction gates, compute A_1/B_1 and let its size be $s_1 \coloneqq \max(\mathrm{size}_{\mathbb{F}}(A_1), \mathrm{size}_{\mathbb{F}}(B_1)) \le \mathrm{poly}(d_u, s_0)$. Let $t = \log d_u + 1$, then Newton iteration gives an easy recurrence on the size $s_t \le c + s_{t-1} + \mathrm{poly}(d_u, s_0)$, where c is a small constant. Solving the recurrence gives $s_t \le \mathrm{poly}(d_u, s_0) \le \mathrm{poly}(d_u, \mathrm{size}(\tilde{f}))$. Finally, eliminate the division with respect to $\boldsymbol{x}, y$ variables using Lemma 3.6 to obtain the circuit computing $\tilde{g}_i$ of size at most $\mathrm{poly}(d_u, \mathrm{size}(\tilde{f}))$. The upper bound is preserved after the inverse transformation τ^{-1}.

Order analysis. Let the ε degree in A_1/B_1 be denoted by $d_1 \coloneqq \max(\deg_\varepsilon(A_1), \deg_\varepsilon(B_1))$. Observe that in each iteration the degree blows-up by a factor of d_u because of homogenisation in preprocessing. Thus, we get the recurrence $d_t \le d_u \cdot d_{t-1}$, solving which gives $d_t \le (d_u)^{\log d_u} \le 2^{\mathrm{poly}(d_u)}$. Then to obtain g_i, set $M = d_t \le 2^{\mathrm{poly}(d_u)}$. □

We are now ready to prove Lemma 4.15. For two arbitrary polynomials u and h, let $\mathrm{size}(u \,|\, h)$ denote the size of the circuit that computes u given h for free. The definition can be extended to $\overline{\mathrm{size}}(u \,|\, h)$ naturally.

Proof of Lemma 4.15. Using the map τ defined earlier, Lemma 4.16 proves that $\tau(u) = k' \cdot$

$\prod_{i \in [d_u]} (y - h_i)^{c_i}$ where $h_i \in \mathbb{F}_q[[x_1, \ldots, x_n]]$ and $d_u := \deg(u)$. Suppose

$$H := k' \cdot \prod_{i \in [d_u]} \left(y - \mathrm{Hom}_{\leq d_u}(h_i)\right)^{c_i} ,$$

then observe that $\tau(u) \equiv H \bmod \langle \boldsymbol{x} \rangle^{d_u+1}$. The idea is to show that H can be easily approximated, hence the factor can be obtained accurately by eliminating division.

It is easy to verify that

$$\overline{\mathrm{size}}(H) \leq \overline{\mathrm{size}}\left(H \mid \mathrm{Hom}_{\leq d_u}(h_i)\right) + \overline{\mathrm{size}}\left(\mathrm{Hom}_{\leq d_u}(h_i)\right) ,$$

see [Bü04, Lemma 2.3(3)]. Since $d_u \leq s$, Proposition 4.17 proves that $\overline{\mathrm{size}}\left(\mathrm{Hom}_{\leq d_u}(h_i)\right)$ is at most $\mathrm{poly}(s)$. Then, clearly $\overline{\mathrm{size}}(H) \leq \mathrm{poly}(s)$. Suppose G approximates H, in the usual sense. Eliminate division in $G \bmod \langle \boldsymbol{x} \rangle^{d_u+1}$ using Lemma 3.6 to obtain the approximation of $\tau(u)$, moreover almost immediately we get that $\overline{\mathrm{size}}(\tau(u)) \leq \mathrm{poly}(s, d_u)$. Since shifting and scaling do not change the size complexity, apply the inverse transformation to conclude that $u \in \overline{\mathsf{VP}}_{\varepsilon}$. ☐

5 VNP is factor closed

We state the three technical lemmas that help us prove Theorem 2, 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 5.1 (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 5.2 (Coprime factors)**.** *Let $f \in \mathbb{F}_q[x_1, \ldots, x_n]$ be a polynomial in* VNP*. If there are coprime 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 5.3 (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 5.1 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 5.2 to shows that u_2 is in VNP, and therefore u_1 is in VNP. We finish the proof by using Lemma 5.1 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 2 formally, which proves the closure of VNP under factoring over all fields.

Theorem 2 (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, [CKS19b, Theorem 2.8] proved 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 $v \in \mathbb{F}_q[x_1, \ldots, x_n]$ satisfying $f =: u^e v$. If $v = 1$, then Lemma 5.3 proves that u is in VNP.

If u and v are coprime, then we conclude the proof using Lemma 5.2 and Lemma 5.3.

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 5.2 and Lemma 5.3 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 3.9); this concludes the proof of u being in VNP. □

5.1 Factoring prime powers or Valiant's converse

To prove Lemma 5.1, 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 5.4 (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 $(\mathrm{poly}(s), \mathrm{poly}(s))$ [8]. That is, there is a polynomial $t_{\boldsymbol{e}}$ over a finite field extension $\mathbb{F}_{q'}$, $q' \leq \mathrm{poly}(s)$, such that

[8] The same conclusion can be made from VNP closure properties stated in Lemma 3.9.

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

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 $\phi_f \in \#\mathsf{P}/\mathsf{poly}$ such that $\phi_f(\boldsymbol{e}) = \langle c_{\boldsymbol{e}}\rangle$. □

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 5.5 (Frobenius Homomorphism)**.** *Let* R *be a commutative ring of characteristic* p*. Define a map* $\rho : \mathsf{R} \to \mathsf{R}$ *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 5.1. Given that $f = u^{p^i}$, let $u =: \sum_{\boldsymbol{a}\in L} c_{\boldsymbol{a}}\boldsymbol{x}^{\boldsymbol{a}}$, where the *support* L represents the set of exponent vectors associated to u. Essentially, Claim 5.5 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 5.4 guarantees a $\#\mathsf{P}/\mathsf{poly}$ function ϕ_f such that the following congruence, in the finite field $\mathbb{F}_q$, is true for all $\boldsymbol{a}\in L$:

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

In Lemma 3.3 it was proved that $\#\mathsf{P}/\mathsf{poly}$ functions are closed under repeated-squaring, hence we conclude that $\phi_u \in \#\mathsf{P}/\mathsf{poly}$. Invoking Proposition 2.1 on ϕ_u proves that the factor $u \in \mathsf{VNP}$. □

5.2 Factoring co-prime factors

The proof of Lemma 5.2 adheres to the conventional template of factoring, pioneered by Kaltofen, using Hensel's lifting lemma. We will follow the presentation of [KSS15, ST21, Sud98]. 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 $\alpha := (\alpha_1,\ldots,\alpha_n)\in\mathbb{F}_q^n$. Define a homogeneous *shift* map $\tau_\alpha : \mathbb{F}_q[x_1,\ldots,x_n] \to \mathbb{F}_q[x,x_1,\ldots,x_n]$ 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

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

PIT lemma guarantees that with high probability, a random choice of α ensures $c_{\boldsymbol{e}}$ is a non-zero field element (refer to [SY10, 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 $\mathrm{poly}(\deg(f))$. Since arithmetic operations over K can be efficiently simulated in $\mathbb{F}$ (refer to [Bü00, 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 $\bar{f} \in \mathbb{F}_q[x_1, \ldots, x_n][x, y]$ as follows: $\bar{f}(x,y) := f_\alpha(x, yx_1 + a_1, yx_2 + a_2, \ldots, yx_n + a_n)$, 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 $\bar{u} := u(x, yx_1 + a_1, \ldots, yx_n + a_n)$ is a monic irreducible factor of $\bar{f}$, see [ST21, 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 [DDS21, Section 2.2] and [ST21, Section 3.5].

Claim 5.6 (Initialize Hensel lifting). *Let $f = u \cdot v$ be such that u, v are co-prime polynomials. Then the associated* univ*ariate 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 $(\bmod\ y)$. For a detailed discussion on the specific monic version of the Lifting lemma required for our proof, we encourage the readers to refer [KSS15, Lemma 3.4]. For the rest of the section we assume $\mathbb{K} := \mathbb{F}_q[x_1, \ldots, x_n]$ as the base ring of the bivariate polynomials in x, y.

Lemma 5.7 (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 $a_0 u_0 + b_0 v_0 \equiv 1 \bmod y$. 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* $a_k u_k + b_k v_k \equiv 1 \bmod y^{2^k}$ *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* $\bmod y^{2^k}$*.*

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

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

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:

$$u_k := u_{k-1} + y^{2^{k-1}} r_k \tag{5.8}$$

$$v_k := \bar{v}_k \left(1 + y^{2^{k-1}} q_k\right)\,. \tag{5.9}$$

It is easy to verify that they are the valid lifts as per Lemma 5.7. Refer [KSS15, 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 (mod 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 $u_t \equiv u \mod y^{2^t}$. Since the lifted factors are unique, u can be obtained from u_t by truncating it to $\deg_y(u)$. Given that $f \in \mathsf{VNP}$, the factor $u \in \mathsf{VNP}$ can be proved using the following technical lemma. It proves that given the coefficients of polynomial f in variables $x_1, \ldots, x_n$, there is a small circuit which computes the lifted factor u.

Lemma 5.10 (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* $f =: \sum_{\boldsymbol{e}_i \in L} c_{\boldsymbol{e}_i}(x_1, \ldots, x_n) \cdot x^{e_{i1}} y^{e_{i2}}$ *.*

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* $\mathrm{Hom}_{\leq d}(u_t)$[9]*. 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.

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

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 5.7. Let the size of all the circuits be at most $s_{t-1} := \text{poly}(d, \beta^{t-1})$. Together with C_f, the difference polynomial m_k can be easily computed in size $s_f + O(s_{t-1})$ [10]. Then observe that $\text{size}(\bar{u}_t)$ and $\text{size}(\bar{v}_t)$ 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 $\text{poly}(d)$ size (refer [KSS15, Lemma 2.8] and [vzGG13, 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 5.8 and 5.9. Overall, the size of the lifted polynomials grows by a constant factor and, hence, the overall size of both the circuits is at most $s_t := s_f + O(s_{t-1}) + \text{poly}(d) + O(\beta) \leq \text{poly}(d, \beta^t)$. 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 using Lemma 3.5, to obtain the desired circuit which computes $\text{Hom}_{\leq d}(u_t)$. 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 5.2.

Lemma 5.2 (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 of Lemma 5.2. Assume that $f \in \mathbb{K}[x, y]$ after all the necessary invertible transformations discussed earlier in the section to apply Lemma 5.7. Let 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 5.10 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 $\text{poly}(\deg(f))$ and degree is at most $O(\deg(f))$.

Since $f \in \mathsf{VNP}$, Lemma 3.9(2) shows that the coefficients $c_{\boldsymbol{e}_i} \in \mathsf{VNP}$. Moreover, Lemma 3.9(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 for 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 problem of proving containment remains open for large fields such as $\mathbb{Q}$, due to the possibility of *doubly*-exponential large integers appearing.

As an application of our debordering result, we advance partially towards proving the factor conjecture [Bü00, Conjecture 8.3], by showing that low-degree 'separable' factors of small size

[10]For notations, refer to the discussion proceeding Lemma 5.7.

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 improves the resolution of Bürgisser's conjecture [Bü00, Conjecture 2.1], [CKS19b].

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 [Sax09, Sax14] and also [SY10, Chapter 4]). It is studied in two well known settings: *Blackbox* and *Whitebox*. The former allows only evaluations, while the latter allows one 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, if g approximates a nonzero 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 the blackbox border PIT, refer [FS18, DDS21].

The arbitrarily large complexity of the 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 power of ε-polynomials, and therefore we make the whitebox setting meaningful. It is worthwhile to investigate whitebox PIT on presentable border classes; for instance, one can begin by studying presentable border of constant depth circuits.

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 [Sax23] 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. This work was completed while P.D. was fulfilling the dissertation requirements at the Department of Computer Science and Engineering, IIT Kanpur.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity*. Cambridge University Press, Cambridge, 2009.

[AGS19] Manindra Agrawal, Sumanta Ghosh, and Nitin Saxena. Bootstrapping variables in algebraic circuits. *Proc. Natl. Acad. Sci. USA*, 116(17):8107–8118, 2019.

[And20] Robert Andrews. Algebraic hardness versus randomness in low characteristic. In *35th Computational Complexity Conference*, volume 169 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 37, 32. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2020.

[AW16] Eric Allender and Fengming Wang. On the power of algebraic branching programs of width two. *Comput. Complexity*, 25(1):217–253, 2016.

[BCRL79] Dario Bini, Milvio Capovani, Francesco Romani, and Grazia Lotti. $\mathrm{O}(\mathrm{n}^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication. *Information Processing Letters*, 8(5):234–235, 1979.

[BCS97] Peter Bürgisser, Michael Clausen, and Mohammad Amin Shokrollahi. *Algebraic complexity theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1997.

[BI18] Markus Bläser and Christian Ikenmeyer. Introduction to geometric complexity theory. Lecture Notes, 2018.

[BIM+20] Markus Bläser, Christian Ikenmeyer, Meena Mahajan, Anurag Pandey, and Nitin Saurabh. Algebraic Branching Programs, Border Complexity, and Tangent Spaces. In *Proceedings of the 35th Annual Computational Complexity Conference (CCC 2020)*, volume 169 of *LIPIcs*, pages 21:1–21:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[Bin80] D. Bini. Relations between exact and approximate bilinear algorithms. Applications. *Calcolo. A Quarterly on Numerical Analysis and Theory of Computation*, 17(1):87–97, 1980.

[BIZ18] Karl Bringmann, Christian Ikenmeyer, and Jeroen Zuiddam. On algebraic branching programs of small width. *J. ACM*, 65(5):Art. 32, 29, 2018.

[BOC92] Michael Ben-Or and Richard Cleve. Computing algebraic formulas using a constant number of registers. *SIAM J. Comput.*, 21(1):54–58, 1992.

[BS21] Pranav Bisht and Nitin Saxena. Blackbox identity testing for sum of special ROABPs and its border class. *Comput. Complexity*, 30(1):Paper No. 8, 48, 2021.

[BSV20] Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. Deterministic factorization of sparse polynomials with bounded individual degree. *J. ACM*, 67(2):Art. 8, 28, 2020.

[Bü00] Peter Bürgisser. *Completeness and reduction in algebraic complexity theory*, volume 7 of *Algorithms and computation in mathematics*. Springer-Verlag, 2000.

[Bü04] Peter Bürgisser. The complexity of factors of multivariate polynomials. *Foundations of Computational Mathematics*, 4(4):369–396, 2004. Preliminary version in the *42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2001)*.

[Bü20] Peter Bürgisser. Correction to: The complexity of factors of multivariate polynomials. *Foundations of Computational Mathematics*, 20(6):1667–1668, 2020.

[CGJ+18] Mahdi Cheraghchi, Elena Grigorescu, Brendan Juba, Karl Wimmer, and Ning Xie. $\mathrm{AC}^0 \circ \mathrm{MOD}_2$ lower bounds for the Boolean inner product. *J. Comput. System Sci.*, 97:45–59, 2018.

[CKS19a] Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Closure of VP under taking factors: a short and simple proof, 2019. Pre-print available at `arXiv:1903.02366`.

[CKS19b] Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Closure results for polynomial factorization. *Theory of Computing. An Open Access Journal*, 15:Paper No. 13, 34, 2019. Preliminary version in the *33rd Annual Computational Complexity Conference (CCC 2018)*.

[CKW10] Xi Chen, Neeraj Kayal, and Avi Wigderson. Partial derivatives in arithmetic complexity and beyond. *Found. Trends Theor. Comput. Sci.*, 6(1-2):front matter, 1–138, 2010.

[CL22] Prasad Chaugule and Nutan Limaye. On the closures of monotone algebraic classes and variants of the determinant. In *LATIN 2022: theoretical informatics*, volume 13568 of *Lecture Notes in Comput. Sci.*, pages 610–625. Springer, 2022.

[CR88] Benny Chor and Ronald L. Rivest. A knapsack-type public key cryptosystem based on arithmetic in finite fields. *IEEE Trans. Inform. Theory*, 34(5, part 1):901–909, 1988.

[CW90] Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251–280, 1990.

[DDS21] Pranjal Dutta, Prateek Dwivedi, and Nitin Saxena. Demystifying the border of depth-3 algebraic circuits. In *Proceedings of the 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2021)*, pages 92–103. IEEE, 2021.

[DS22] Pranjal Dutta and Nitin Saxena. 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)*, pages 200–211. IEEE, 2022.

[DSS22] Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. Discovering the Roots: Uniform Closure Results for Algebraic Classes Under Factoring. *J. ACM*, 69(3):18:1–18:39, 2022.

[DSY10] Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM J. Comput.*, 39(4):1279–1293, 2009/10.

[For14] Michael A. Forbes. *Polynomial identity testing of read-once oblivious algebraic branching programs*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2014.

[For16] Michael Forbes. Some Concrete Questions on the Border Complexity of Polynomials, 2016. Talk at the Workshop on Algebraic Complexity Theory (WACT) 2016 in Tel Aviv.

[FS15] Michael Forbes and Amir Shpilka. Complexity Theory Column 88: Challenges in Polynomial Factorization1. *SIGACT News*, 46(4):32–49, Dec 2015.

[FS18] Michael A. Forbes and Amir Shpilka. 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, 2018.

[FSTW21] Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. Proof complexity lower bounds from algebraic circuit complexity. *Theory Comput.*, 17:Paper No. 10, 88, 2021.

[GKSS22] Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. Derandomization from algebraic hardness. *SIAM J. Comput.*, 51(2):315–335, 2022.

[GMQ16] Joshua A. Grochow, Ketan D. Mulmuley, and Youming Qiao. Boundaries of VP and VNP. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP 2016)*, volume 55 of *LIPIcs*, pages 34:1–34:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[Gre16] Bruno Grenet. Bounded-degree factors of lacunary multivariate polynomials. *J. Symbolic Comput.*, 75:171–192, 2016.

[GS99] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Trans. Inform. Theory*, 45(6):1757–1767, 1999.

[GS11] S. B. Gashkov and I. S. Sergeev. Complexity of computations in finite fields. *Fundamentalnaya i Prikladnaya Matematika*, 17(4):95–131, December 2011.

[Jan11] Maurice J. Jansen. Extracting Roots of Arithmetic Circuits by Adapting Numerical Methods. In *Innovations in Computer Science - ICS*, pages 87–100. Tsinghua University Press, 2011.

[Kal85] Erich Kaltofen. Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization. *SIAM J. Comput.*, 14(2):469–489, 1985.

[Kal86] Erich L. Kaltofen. Uniform Closure Properties of P-Computable Functions. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC 1986)*, pages 330–337. ACM, 1986.

[Kal87] Erich Kaltofen. 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)*, page 443–452. Association for Computing Machinery, 1987.

[Kal89] Erich Kaltofen. Factorization of Polynomials Given by Straight-Line Programs. *Adv. Comput. Res.*, 5:375–412, 1989.

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

[KK08] Erich Kaltofen and Pascal Koiran. Expressing a fraction of two determinants as a determinant. In *Proceedings of the 2008 International Symposium on Symbolic and Algebraic Computation (ISSAC 2008)*, pages 141–146. ACM, 2008.

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

[KS19] Mrinal Kumar and Ramprasad Saptharishi. Hardness-Randomness Tradeoffs for Algebraic Computation. *Bull. EATCS*, 129, 2019.

[KSS15] Swastik Kopparty, Shubhangi Saraf, and Amir Shpilka. Equivalence of polynomial identity testing and polynomial factorization. *Computational Complexity*, 24(2):295–331, 2015.

[KST19] Mrinal Kumar, Ramprasad Saptharishi, and Anamay Tengse. Near-optimal bootstrapping of hitting sets for algebraic circuits. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 639–646. SIAM, 2019.

[KT90] Erich Kaltofen and Barry M. Trager. Computing with polynomials given by black boxes for their evaluations: greatest common divisors, factorization, separation of numerators and denominators. *J. Symbolic Comput.*, 9(3):301–320, 1990.

[Kum20] Mrinal Kumar. On the Power of Border of Depth-3 Arithmetic Circuits. *ACM Trans. Comput. Theory*, 12(1):5:1–5:8, 2020.

[Lan17] J. M. Landsberg. *Geometry and Complexity Theory*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2017.

[Len99] H. W. Lenstra, Jr. Finding small degree factors of lacunary polynomials. In *Number theory in progress, Vol. 1 (Zakopane-Kościelisko, 1997)*, pages 267–276. de Gruyter, 1999.

[LO15] Joseph M. Landsberg and Giorgio Ottaviani. New lower bounds for the border rank of matrix multiplication. *Theory of Computing. An Open Access Journal*, 11:285–298, 2015.

[Mah14] Meena Mahajan. *Algebraic Complexity Classes*, pages 51–75. Springer International Publishing, 2014.

[Mal03] Guillaume Malod. *Polynômes et coefficients. (Polynomials and coefficients)*. PhD thesis, Claude Bernard University Lyon , France, 2003.

[MP08] Guillaume Malod and Natacha Portier. Characterizing Valiant's algebraic complexity classes. *J. Complex.*, 24(1):16–38, 2008.

[MP13] Gary L. Mullen and Daniel Panario. Introduction to finite fields : Basic properties of finite fields. In *Handbook of Finite Fields*, Discrete mathematics and its applications, pages 13–31. CRC Press, 2013.

[MS01] Ketan D. Mulmuley and Milind Sohoni. Geometric complexity theory. I. An approach to the P vs. NP and related problems. *Siam Journal On Computing*, 31(2):496–526, 2001.

[MS08] Ketan D. Mulmuley and Milind Sohoni. Geometric complexity theory. II. Towards explicit obstructions for embeddings among class varieties. *SIAM J. Comput.*, 38(3):1175–1206, 2008.

[Mul11] Ketan D. Mulmuley. On P vs. NP and geometric complexity theory. *J. ACM*, 58(2):Art. 5, 26, 2011.

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

[Mum76] David Mumford. *Algebraic geometry. I*. Grundlehren der Mathematischen Wissenschaften. Springer-Verlag, 1976. Complex projective varieties.

[Oli16] Rafael Oliveira. Factors of low individual degree polynomials. *Comput. Complexity*, 25(2):507–561, 2016.

[Oli20] Rafael Oliveira. 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)*, pages 396–401. ACM, 2020.

[Pap94] Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley Publishing Company, Reading, MA, 1994.

[Pé04] Sylvain Périfel. *Polynômes donnés par des circuits algébriques et généralisation du modèle de Valiant*. École Normal Supérieure de Lyon, France, 2004. Master's Thesis.

[Reg02] Kenneth W. Regan. Understanding the Mulmuley-Sohoni approach to P vs. NP. *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS*, 78:86–99, 2002.

[Sap15] Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Github Survey, 2015.

[Sax09] Nitin Saxena. Progress on polynomial identity testing. *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS*, 99:49–79, 2009.

[Sax14] Nitin Saxena. Progress on polynomial identity testing-II. In *Perspectives in computational complexity*, volume 26 of *Progr. Comput. Sci. Appl. Logic*, pages 131–146. Birkhäuser/Springer, 2014.

[Sax23] Nitin Saxena. Closure of algebraic classes under factoring, 2023. Talk at Recent Trends in Computer Algebra (2023) in Institut Henri Poincaré, Paris.

[Sho09] Victor Shoup. *A computational introduction to number theory and algebra*. Cambridge University Press, second edition, 2009.

[SK12] Jayalal Sharma and Dinesh K. Advanced Complexity Theory. Lecture Notes, 2012.

[ST21] Amit Sinhababu and Thomas Thierauf. Factorization of Polynomials Given by Arithmetic Branching Programs. *Comput. Complex.*, 30(2):15, 2021.

[Str74] Volker Strassen. Polynomials with rational coefficients which are hard to compute. *Siam Journal On Computing*, 3:128–149, 1974.

[Sud97] Madhu Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *J. Complexity*, 13(1):180–193, 1997.

[Sud98] Madhu Sudan. Algebra and Computation. Lecture Notes, 1998.

[SV10] Amir Shpilka and Ilya Volkovich. On the relation between polynomial identity testing and finding variable disjoint factors. In *Automata, languages and programming. Part I*, volume 6198 of *Lecture Notes in Comput. Sci.*, pages 408–419. Springer, 2010.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic Circuits: A Survey of Recent Results and Open Questions. *Found. Trends Theor. Comput. Sci.*, 5(3–4):207–388, 2010.

[Val79] Leslie G. Valiant. Completeness Classes in Algebra. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing (STOC 1979)*, pages 249–261. ACM, 1979.

[Val82] L. G. Valiant. Reducibility by algebraic projections. In *Logic and algorithmic*, volume 30 of *Monogr. Enseign. Math.*, pages 365–380. Univ. Genève, Geneva, 1982.

[vzG84] Joachim von zur Gathen. Hensel and Newton methods in valuation rings. *Math. Comp.*, 42(166):637–661, 1984.

[vzGG13] Joachim von zur Gathen and Jürgen Gerhard. *Modern computer algebra*. Cambridge University Press, Cambridge, third edition, 2013.

[vzGK85] Joachim von zur Gathen and Erich L. Kaltofen. Factoring Sparse Multivariate Polynomials. *J. Comput. Syst. Sci.*, 31(2):265–287, 1985.