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 (\(\text{VP}\)) is quite popular due to the Geometric Complexity Theory (GCT) approach to \(\text{P} \neq \text{NP}\) 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 \(C(\varepsilon, \mathbf{x})\) cannot be compactly presented in reality, as the limit parameter \(\varepsilon\) 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 \(F_q(\varepsilon)\), that it can generate by the ring operations on \(\{\varepsilon, 1/\varepsilon\} \cup F_q\) within poly-size circuit. This model is more expressive than \(\text{VP}\) as it affords exponential-degree in \(\varepsilon\); and analogous to the usual border, we define new border classes called \(\text{VP}_\varepsilon\) and \(\text{VNP}_\varepsilon\). We prove that both these (now called presentable border) classes lie in \(\text{VNP}\). Such a ‘debordering’ result is not known for the classical border classes \(\text{VP}\) resp. \(\text{VNP}\). We pose \(\text{VP}_\varepsilon = \text{VP}\) 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 completely solve (Conj.2.1 in Bürgisser 2000; Chou, Kumar, Solomon CCC 2018):

1. Each poly-degree irreducible factor, with multiplicity coprime to field characteristic, of a poly-size circuit (of possibly exponential-degree), is in \(\text{VNP}\).

2. For all finite fields, and all factors, \(\text{VNP}\) is closed under factoring. Consequently, factors of \(\text{VP}\) are always in \(\text{VNP}\). The prime characteristic cases were open before due to the inseparability obstruction (i.e. when the multiplicity is not coprime to \(q\)).

We also provide analogous theorems of explicitness over characteristic zero fields (eg. number fields).

∗CSE, IIT Kanpur, bhargav@cse.iitk.ac.in
†CSE, IIT Kanpur, pdwivedi@cse.iitk.ac.in
‡CSE, IIT Kanpur, nitin@cse.iitk.ac.in
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 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 $F$, and a designated output node. The circuit computes an $n$-variate polynomial $f(x) \in F[x_1, \ldots, x_n]$ in a natural bottom-up way.

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 $F$) computing the polynomial $f$ by $\text{size}_F(f)$. 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 $\text{VP} \neq \text{VNP}$ conjecture. The class $\text{VP}$ (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 $\text{NP}$ using an exponential sum of $\text{VP}$ polynomials. More formally,

1 Computing a polynomial always refers to computing a family of polynomials $\{f_n\}$, one for each $n \in \mathbb{N}$. 

**Definition 1.1 (Valiant’s NP).** The class $\text{VNP}$ 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 $\text{VP}$ with $m = \text{poly}(n)$ and

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

We call $y_1, \ldots, y_m$ the witness (or hypercube) variables and $g(x, y)$ as the verifier circuit. It is straightforward to see that $\text{VP} \subseteq \text{VNP}$, 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 (and the one we will use) 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 $g(x, \varepsilon) = \varepsilon^M f(x) + \varepsilon^{M+1}Q(x, \varepsilon)$, for some $Q \in \mathbb{F}[\varepsilon][x_1, \ldots, x_n]$. The border size of $f$ denoted $\overline{\text{size}}(f)$, is defined as $\text{size}_{\mathbb{F}[\varepsilon]}(g)$, the size of the polynomial $g$ over the ring $\mathbb{F}[\varepsilon]$ (instead of being over the constants $\mathbb{F}$).

Note that $\lim_{\varepsilon \to 0} \varepsilon^{-M} g(x, \varepsilon) = f(x)$. Furthermore, arbitrary polynomials in $\varepsilon$ are treated as ‘free constants’ in the circuit of $g$. Alternately, we can also consider the approximating polynomial $g$ over the rational function field $\mathbb{F}(\varepsilon)$ (as done in our paper abstract) and aim for an approximation of the form $g' = f + \varepsilon Q$, with the effect of $\lim_{\varepsilon \to 0} g' = f$. It is not hard to see via scaling arguments ($g' := \varepsilon^{-M} g$) 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 $\text{VP}$, called $\overline{\text{VP}}$ as the set of $\text{poly}(n)$-degree polynomials whose border size is bounded by $\text{poly}(n)$. Clearly, $\text{VP} \subseteq \overline{\text{VP}}$. In an ambitious program to resolve the $P \neq \text{NP}$ question using methods from algebraic geometry and representation theory, Mulmuley and Sohoni [MS01] strengthened Valiant’s conjecture by postulating that $\text{VNP}$ is not contained in $\overline{\text{VP}}$. 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].

\footnote{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] (also see [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 $\text{VP}$ 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 $\text{poly}(e, \deg(u), \text{size}(f))$. One might expect, for exponential-degree $f$, that the size of $u$ depends only on its degree and the size of $f$, and that the dependence on multiplicity $e$ can be completely removed. In other words, we expect that any poly($n$)-degree factor of a poly($n$)-size circuit (with no restrictions on degree) is in $\text{VP}$. This is known as the Factor Conjecture [Bü00, Conjecture 8.3]. In his work, Bürgisser [Bü04] showed that for border complexity, the factor conjecture is indeed true – the factor $u$ above, is in $\overline{\text{VP}}$. This makes factor conjecture an important stepping-stone towards understanding algebraic computation. Our work will build on this theme.

1.2 Our goal: To make border presentable

The notion of approximation in Valiant’s framework arose at the same time in different contexts. This suggests that it is indeed very natural. But a basic question, made even more pertinent by the discussion above, that remains open to this day is whether approximation bestows more computational power, or in other words, whether $\text{VP} \supseteq \overline{\text{VP}}$ [Bü04, Problem 4.3]. In a recent work [DDS21] asked a more general question, which they called de-bordering. Given a polynomial $f \in \mathbb{C}$ 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 $\varepsilon$ of arbitrary complexity to be used as free constants. This arbitrariness makes the definition of approximation inherently existential. In fact, we do not even know whether $\overline{\text{VP}}$ is contained in $\text{VNP}$.

As a way of making approximation more constructive, while retaining its essence, in this work we propose and study a natural restriction on the definition of approximation, that we call presentability. The presentable class $\overline{\text{VP}}_\varepsilon$ is the same as $\overline{\text{VP}}$ but with the additional condition that all the polynomials in $\varepsilon$ used as ‘constants’ in the approximating circuit $g(x, \varepsilon)$, have polynomial-size circuits themselves (see Definition 3.12).

There has previously been an attempt via ‘degenerations’ [GMQ16] to identify a subclass of $\overline{\text{VP}}$ that is explicit. In what they term $p$-definable one-parameter degeneration, the authors restrict the coefficients of the $\varepsilon$-polynomials to be generated using circuits in $\text{VP}$. Our presentable border is a more natural version of $\overline{\text{VP}}$ and cannot be obtained as a $p$-definable degeneration of $\text{VP}$, making our notion incomparable to the concept of degeneration as studied in [GMQ16]. We can extend our concept of presentable border to $\overline{\text{VNP}}_\varepsilon$ over any field $\mathbb{F}$. 

4
Definition 1.2 (Presentable $\overline{\text{VNP}}$). The presentable border class $\overline{\text{VNP}}_\varepsilon$, over $\mathbb{F}$, is defined as the set of polynomials $f \in \mathbb{F}[x_1, \ldots, x_n]$ such that there is an approximating polynomial $g \in \mathbb{F}[\varepsilon][x_1, \ldots, x_n]$ expressing

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

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 $m, \deg_{x,y}(h)$ and $\text{size}_\mathbb{F}(h)$ all bounded by poly$(n)$, satisfying a hypercube-sum expression

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

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

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

Remark. The theorem continues to hold over number fields if all rational numbers that appear in the computation have polynomial bit complexity. Refer to Theorems 3 and 4 for the formal statements.

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

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 poly$(n)$-degree factor of a poly$(n)$-size circuit is in $\overline{\text{VP}}$. We observe that it is in fact in $\overline{\text{VP}}_\varepsilon$, and thus by Theorem 1 in $\text{VNP}$.

Corollary 1.3 (Debordering factors). Let $f_n$ be a $n$-variate polynomial family over a finite field that has a poly$(n)$-degree irreducible factor $u_n$ of multiplicity co-prime to the characteristic of the field. If $\text{size}(f_n)$ is poly$(n)$, then $u_n$ is in $\text{VNP}$.

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. The result holds over $\mathbb{Q}$ and its extensions, up to polynomial bit complexity.
3. We get an *explicitness* $(\text{VNP})$ 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 $\text{VNP}$ 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 $\text{VNP}$ closure under factoring holds over finite fields as well, thus settling Bürgisser’s conjecture completely.

**Theorem 2** (Factor closure). *Over any finite field, the class $\text{VNP}$ is closed under factorization.*

*Remark.* As a corollary of the above theorem, we find that over finite fields, the factors of polynomials in $\text{VP}$ are in $\text{VNP}$. This partially answers the question [Bü00, Problem 2.1] whether $\text{VP}$ is closed under taking factors over fields of positive characteristic. Recall that over fields of characteristic zero, we already know this to be true from the works of Kaltofen; but those methods fail in finite fields.

### 2 Proof outline

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

#### 2.1 Efficacy of presentable border

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

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

says very little about the complexity of the $\varepsilon$-constants involved, which could be huge. A natural idea to isolate $f$ from the above expression is via interpolation on the $\varepsilon$ variable. This seems hard to do as apriori, the degree of $\varepsilon$ 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 $\text{size}(f) := \text{size}_\varepsilon(f)$, the *border size* of the polynomial $f$. Therefore, moving to presentable border classes $\overline{\text{VP}}_\varepsilon$ and $\overline{\text{VNP}}_\varepsilon$ does not lead to any $\varepsilon$-degree loss, since they allow for an exponential degree in $\varepsilon$. But unless one can show a *polynomial* bound on the order of approximation \(^3\), interpolation seems to give a bound of the form $\text{size}(f) \leq \exp(\text{size}(f))$.

**Known debordering results.** Incidentally, the known de-bordering results for restricted models of computation seldom use interpolation. In his talk [For16], Forbes remarked that Nisan’s characterization implies the closure of ROABPs or equivalently non-commutative ABPs (see [For14, \(^4\)See [BIZ18, Corollary 3.10] for an example of debordering through interpolation when a related measure of approximation called ’error degree’ is *polynomially* bounded.]\(^3\))
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. Using elementary but clever matrix identities, [BIZ18] showed that the closure of width-2 algebraic branching programs is the same as the closure of general formulas. Together 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 \( P_k \) and \( P_{k+1} \). 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 \( \text{VP}_\varepsilon \) and \( \text{VNP}_\varepsilon \).

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 \( \text{VNP} \subseteq \text{VNP}_\varepsilon \) by definition. For the other direction, to show the containment in \( \text{VNP} \), 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 \( \text{VNP} \) 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 \( f = \sum_{e} c_e x^e \) be a polynomial in \( n \) variables of degree \( \text{poly}(n) \) over a field \( \mathbb{F} \). Suppose that there exists a string function \( \phi : \{0,1\}^* \rightarrow \{0,1\}^* \) in \( \text{#P} / \text{poly} \) such that \( \phi(\langle e \rangle) = \langle c_e \rangle \). Then, the polynomial \( f \) is in \( \text{VNP} \) over the field \( \mathbb{F} \).

**Remark.** Unlike the usual definition of \( \text{#P} \) which consists of functions mapping \( \{0,1\}^* \) to \( \mathbb{N} \), we find it more convenient to consider functions that output binary strings (Definition B.1). 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 \( O(a \log p) \).

Consider now a polynomial \( f = \sum_{e} c_e x^e \in \text{VNP}_\varepsilon \) (as in Definition 1.2) over the finite field \( \mathbb{F}_q \). We would like to show that the coefficient function \( \phi : \langle e \rangle \mapsto \langle c_e \rangle \) is in \( \text{#P} / \text{poly} \). We have access
to $f$ only using the approximating polynomial $g$

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

which is of the following hypercube-sum form

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

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

We will extract the coefficient of $\varepsilon^M x^e$ 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 $c_e$ can be obtained as a hypercube sum of an exponential degree algebraic circuit of polynomial size (Lemma 3.1) 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 $\text{VNP}$, mainly because the $\varepsilon$-constants might truly require exponential size circuits. Working with $\text{VNP}_\varepsilon$ lets us keep the circuit size small while retaining the exponentially large degree of $\varepsilon$.

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 $\omega$ of an exponentially large field $\mathbb{F}_{q'}$ to realize the hypercube points.

Using finite field arithmetic and the closure of the Boolean class $\#P$ under exponential sums, we move from the algebraic world to the Boolean one (Lemma 3.2). Thus, we show that the algebraic circuit above (from Lemma 3.1) can be simulated by a (multi-output) Boolean circuit of polynomial size; furthermore, the hypercube sum computing the coefficient function is demonstrated in $\#P/\text{poly}$. Valiant’s criterion (Proposition 2.1) now implies that the polynomial $f$ is indeed in $\text{VNP}$.

To prove explicitness over $\mathbb{Q}$, we reduce the problem to the case of finite fields. But in order to do this transfer, we need to restrict the numbers appearing in the computation or else, one has to deal with enormous numbers with arbitrary bit complexity. The main idea is to choose a ‘good prime’ and work over the field $\mathbb{F}_p$ and later reconstruct the rational numbers. This involves
a restriction on the coefficients of the polynomial as well as the computational model. We point
the reader to Section 3.3 for more details and extensions to number fields.

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 \(^4\) 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.

The proofs and techniques introduced in these works have evolved to provide applications
in various areas of computer science, eg. 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 impor-
tant 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 4.3)
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

\(^4\)There have been many proofs of the original VP closure result itself! See [Bü00, KSS15, Oli16, CKS19a, DSS22]
for some alternate ones.
contribution is showing that if \( f = u^k \) for some \( k \geq 1 \), then \( u \) is in \( \text{VNP} \) (Lemma 4.2). Using this result, we can then handle all powers (Lemma 4.4).

All previous known techniques fail in the case where the exponent \( e \) is a prime power. 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 \( \text{VNP} \) also has a converse (Lemma 4.5). It was remarked in [MP08, Section 6] that the fact has been observed before in [Pé04], though we could not find a written reference. We give an independent proof for finite fields in this paper by first noting that any coefficient of a \( \text{VNP} \) polynomial can be obtained as a hypercube-sum of evaluations of a \( \text{VP} \) circuit. Next, we use ideas similar to the proof of Theorem 1 to convert the algebraic expression thus obtained to a Boolean \( \#P/\text{poly} \) circuit.

Since \( f \in \text{VNP} \), the inverse of Valiant’s criterion gives us that its coefficient function is in \( \#P/\text{poly} \). We obtain the coefficients of \( u \) by performing an inverse Frobenius transform, which we demonstrate in \( \#P/\text{poly} \). Finally, using Valiant’s criterion in the forward direction, we see that the factor \( u \) is in \( \text{VNP} \).

3 Presentable is explicit: Proof of Theorem 1

In this section we will prove that polynomials in \( \text{VNP}_\varepsilon \) are explicit over finite fields. Later in Section 3.3, we will discuss analogous results over rationals and its extensions.

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 3.1** (Exponential interpolation). Let \( s := \text{poly}(r, \log q) \) and let \( g = \sum_{e} c_e y^e \) be an \( r \)-variate polynomial over \( \mathbb{F}_q \) of degree \( D := \exp(s) \) such that \( g = \sum_{a \in \{0, 1\}^m} h(y, a) \) for some polynomial \( h \) with \( m, \text{size}(h) \leq s \).

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

We will prove the above lemma in Section 3.1. In the subsequent lemma we show that the resulting hypercube sum above can be converted into a boolean function in \( \#P/\text{poly} \). The two lemmas together build up the correct setup to invoke Valiant’s criterion. Recall \( s = \text{poly}(r, \log q) \).

---

5Perifel communicated to us a proof that over \( \mathbb{Q} \), the coefficients of constant-free \( \text{VNP} \) families (see [Mal03]) are in \( \text{GapP}/\text{poly} \).
Lemma 3.2 (Algebraic to boolean complexity). For any exponent vector \( e \in \{0, \ldots, D\}^r \), let the coefficient of \( y^e \) in \( g \in \mathbb{F}_q[y_1, \ldots, y_r] \), denoted by \( c_e \), be computable by a polynomial \( t_e \) over a finite field extension \( \mathbb{F}_{q'} \), \( q' \leq \text{poly}(D) \leq 2^{O(s)} \), as follows:

\[
 c_e = \sum_{b \in \{0,1\}^\ell} t_e(b_1, \ldots, b_\ell),
\]

where \( \ell \) and \( \text{size}(t_e) \) are at most \( \text{poly}(s) \). Then, with \( s \) as the input-size parameter, there exists a function \( \phi_g \) in \( \#P / \text{poly} \) that computes \( \phi_g(\langle e \rangle) = \langle c_e \rangle \).

We will defer the proof of the lemma until Section 3.2. Meanwhile, we will use the technical lemmas to give the complete proof of our first main result.

Proof of Theorem 1. Consider a polynomial (family) \( f = f_n \in \mathbb{F}_q[x_1, \ldots, x_n] \) in \( \overline{\text{VNP}}_\varepsilon \) 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 \( \overline{\text{VNP}}_\varepsilon \) size parameters of \( g \) be \( (s, s) \), where \( s := \text{poly}(n) \) and \( d := \deg_\varepsilon(g) \leq \text{poly}(s) \). 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 3.1 on \( g \), followed by applying Lemma 3.2, gives a \( \#P / \text{poly} \) function \( \phi_g \) which computes the encoding of coefficients of \( g \). The coefficient of a monomial \( x^e \) in \( f \) is the coefficient of \( \varepsilon^M \cdot x^e \) in the approximating polynomial \( g \). Observe that if

\[
 f = \sum_{e \in \{0,\ldots,d\}^n} c_e \cdot x^e,
\]

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

3.1 Exponential interpolation technique: Proof of Lemma 3.1

In this section we will give the proof of Lemma 3.1. 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 \( s = \text{poly}(r + m, \log q) \) and \( q =: p^a \) for prime \( p \). We will induct on the number of variables \( r \).

Consider a positive integer \( k \) such that \( 2^a = D < k < \Theta(D) \), and a primitive root of unity \( \omega \) of order \( k \). We know that \( \omega \in \mathbb{F}_q \) if and only if \( k \) divides \( q - 1 \) (refer \cite{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 \cite[Chapter 8]{vzGG13} and \cite[Exercise 17.24]{Sho09}. 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 $V_\omega := (\omega^{ij})_{0 \leq i,j < k}$ as follows:

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

Vandermonde matrices are invertible if and only if its entries are all distinct. It is clear that $\omega^{-1}$ 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:

$$
c_e = \sum_{j=0}^{k-1} \frac{\omega^{-ej}}{k} \cdot g(\omega^j) = \sum_{a \in \{0,1\}^m} \left( \sum_{j=0}^{k-1} \frac{\omega^{-ej}}{k} \cdot h(\omega^j, a) \right)_{\omega^{-1}}. \quad (3.5)
$$

A circuit that computes the inner sum in Equation 3.5 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 $\omega$ 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 $\text{wt}(k) := \lceil \log_2 k \rceil$ and use it to define a polynomial $\bar{h} \in \mathbb{F}_q[z, z_1, \ldots, z_{\text{wt}(k)}]$ as follows:

$$
\bar{h} := \prod_{i=1}^{\text{wt}(k)} \left( z_i \cdot z_i^{j_i} + (1 - z_i) \cdot 1 \right). \quad (3.6)
$$

Let $j := (j_1, \ldots, j_{\text{wt}(k)})$ be the binary representation of $j$, then it is easy to verify that $\bar{h}(\omega, j) = \omega^j$. Together with $\bar{h}$, Equation 3.5 can be re-written as follows:

$$
c_e = \sum_{a \in \{0,1\}^m} \sum_{j \in \{0,1\}^{\text{wt}(k)}} \frac{1}{k} \cdot \bar{h}(\omega^{-1}, (e)), j) \cdot h(\bar{h}(\omega, j), a) =: \sum_{a \in \{0,1\}^\ell} t_a(a, j),
$$

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

12
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

\[
g = \sum_{i \leq D} g_i(y_2, \ldots, y_n) \cdot y_1^i,
\]

where \( g_i \) is \((r-1)\)-variety polynomial of degree at most \( D \). With respect to the fixed exponent vector \( \mathbf{e} = (e_1, e_2, \ldots, e_r) \in \mathbb{N}^r \), define \( \mathbf{e}^- := (e_2, \ldots, e_r) \in \mathbb{N}^{r-1} \). From the equation above, observe that computing the coefficient \( c_{\mathbf{e}} \) of \( y_1^{e_1} y_2^{e_2} \cdots y_r^{e_r} \) in \( g \) is equivalent to computing the coefficient \( c_{\mathbf{e}^-} \) 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 \( \{\omega^0, \omega^1, \ldots, \omega^{k-1}\} \). The equivalent linear system obtained using the Vandermonde matrix \( V_\omega \) is as follows:

\[
\begin{pmatrix}
1 & 1 & 1 & \ldots & 1 \\
1 & \omega & \omega^2 & \ldots & \omega^{k-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \omega^{k-1} & \omega^{2(k-1)} & \ldots & \omega^{(k-1)^2}
\end{pmatrix}
\begin{pmatrix}
\vdots \\
g_{e_1}
\end{pmatrix}
=
\begin{pmatrix}
\vdots \\
g(\mathbf{e}_1, \mathbf{y})
\end{pmatrix}
\begin{pmatrix}
1 & 1 & 1 & \ldots & 1 \\
1 & \omega & \omega^2 & \ldots & \omega^{k-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \omega^{k-1} & \omega^{2(k-1)} & \ldots & \omega^{(k-1)^2}
\end{pmatrix}_{k \times k}
\begin{pmatrix}
\vdots \\
1 & 1 & 1 & \ldots & 1 \\
1 & \omega & \omega^2 & \ldots & \omega^{k-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \omega^{k-1} & \omega^{2(k-1)} & \ldots & \omega^{(k-1)^2}
\end{pmatrix}_{k \times k}
\end{pmatrix}
\]

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

\[
ge_{e_1} = \sum_{j=0}^{k-1} \frac{\omega^{-e_1 j}}{k} \cdot g(\omega^j, y_2, \ldots, y_r)
= \sum_{a \in \{0,1\}^m} \left( \sum_{j=0}^{k-1} \frac{\omega^{-e_1 j}}{k} \cdot h(\omega^j, y_2, \ldots, y_r, a) \right).
\]

To show that the inner summation has small size circuit, we encode the powers of root of unity using the polynomial \( \tilde{h} \) defined in Equation 3.6. All together, it gives the following compact expression:

\[
ge_{e_1} = \sum_{a \in \{0,1\}^m} \sum_{j \in \{0,1\}^{\text{wt}(k)}} \frac{1}{k} \cdot \tilde{h}(\tilde{h}(\omega^{-1}, \langle e_1 \rangle), j) \cdot h(\omega, j, y_2, \ldots, y_r, a)
\]

where \( \ell := m + \text{wt}(k) \leq O(s) \). Further, size(h_{e_1}) \leq s + 2 \times \text{size}(\tilde{h}) \leq O(s).

Size analysis. We analyse the size of the verifier-circuit of \( c_e \) by unfolding the induction
layers. Since $g_{c_1}$ is now a $(r - 1)$-variate polynomial, using induction hypothesis we get that there is a polynomial $t_e$, that computes the relevant coefficient $c_{e^-}$ as follows:

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

whence we define $s(r) := \text{size}(t_e)$. Building on the insights from Equation 3.8, observe that in each iteration of the interpolation, the verifier is only evaluated and multiplied by the polynomial $\tilde{h}$. This stems from the nature of interpolation, which extracts the coefficients as a linear combination of polynomial evaluations. So, we get a simple recurrence: $s(r) \leq s(r - 1) + 2 \cdot \text{size}(\tilde{h})$, which implies that the final verifier-circuit size $s(r) \leq O(rs)$. Analogously, the witness length increases by $\text{wt}(k)$ in each iteration, hence $\ell(r) \leq m + r \cdot \text{wt}(k) \leq O(rs)$. That concludes the proof of Lemma 3.1.

3.2 Transfer algebraic complexity to boolean: Proof of Lemma 3.2

In this section, we will show that the hypercube-sum of the evaluations of a small-size circuit can be transformed into a $\#\text{P}/\text{poly}$ function, which will prove Lemma 3.2. As described earlier, the proof goes via booleanisation of the algebraic circuit. Recall that $q = p^a$, and for a field element $b \in \mathbb{F}_q$, $\langle b \rangle \in \{0,1\}^a$ denotes the binary encoding of $b$. For a point $b \in \mathbb{F}_q^\ell$, denote $\langle b \rangle := (\langle b_1 \rangle, \ldots, \langle b_\ell \rangle) \in \{0,1\}^{\ell a}$.

**Claim 3.9 (Booleanisation).** Consider a polynomial $t \in \mathbb{F}_q[y_1, \ldots, y_\ell]$ such that $\text{size}(t) \leq s$. There exists an equivalent (multi-output) boolean circuit $T$ of bitsize $\leq s \cdot \text{poly}(\log q)$, such that for all inputs $b \in \mathbb{F}_q^\ell$ we have $T(\langle b \rangle) = \langle t(b) \rangle$.

**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 $\beta$ in $\mathbb{F}_q$ can be split into $\log q$ many gates based on $\langle \beta \rangle$. Therefore 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 $s \cdot \text{poly}(\log q)$ such that for all inputs $b \in \mathbb{F}_q^\ell$ we get $T_i(\langle b \rangle) = \langle C_i(b) \rangle$, for $i \in \{1,2\}$. 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 $\text{poly}(\log q)$ 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 $\langle b_1 \times b_2 \rangle = T_\times(\langle b_1 \rangle, \langle b_2 \rangle)^6$. For

---

6The Boolean encoding and the output of Boolean circuit are compared coordinate-wise.
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 $b \in \mathbb{F}_q^\ell$ we have $T_{12}(\langle b \rangle) = \langle C_{12}(b) \rangle$. Notice that bitsize($T_{12}$) = bitsize($T_1$) + bitsize($T_2$) + max(bitsize($T_+$), bitsize($T_\times$)). Proceeding this way in a level-by-level fashion, we obtain the complete Boolean circuit $T$ which computes $\langle t(b) \rangle$. Finally, for the bitsize claim we observe that every gate is replaced by either $T_+$ or $T_\times$ and thus bitsize($T$) ≤ $s \cdot \max(\text{bitsize}(T_+), \text{bitsize}(T_\times))$ ≤ $s \cdot \text{poly}(\log q)$.

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 for coefficients from Lemma 3.2:

$$c_e = \sum_{b \in \{0,1\}^\ell} t_e(b_1, \ldots, b_\ell)$$

where $\ell$ and size($t_e$) are at most poly($s$). Since $c_e$ and $t_e(b)$ 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 3.2.** Consider the $\mathbb{F}_p$-basis representation of $\mathbb{F}_q$ element $t_e(b) = \sum_{i < a} t_{e,i} \alpha^i$, where $t_{e,i} \in \mathbb{F}_p$. Apply Claim 3.9 to the algebraic circuit that computes $t_e$ to obtain a multi-output equivalent Boolean circuit $T_e$ satisfying $\langle t_e(b) \rangle = T_e(\langle b \rangle)$, for all $b \in \{0,1\}^\ell$. The Boolean circuit $T_e$ computes the encoding of $\mathbb{F}_q$ element as a tuple ($\langle t_{e,0} \rangle, \ldots, \langle t_{e,a-1} \rangle$). Let $T_{e,i}$ denote a sub-circuit of $T_e$ computing the string $\langle t_{e,i} \rangle$.

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

Let the $\mathbb{F}_p$-basis representation of the coefficient be $c_e = \sum_{i < a} c_{e,i} \alpha^i$, where $c_{e,i} \in \mathbb{F}_p$. To design the coefficient function $\phi_g$ that computes the encoding of $c_e$, it suffices to prove that there is a function $\phi_{g,i}(\langle e \rangle)$ in $\#P/\text{poly}$ that computes $\langle c_{e,i} \rangle$, for all $i < a$ (see Claim B.4 and remark of Proposition 2.1). From Equation 3.3, we see that $c_{e,i} = \sum_{b \in \{0,1\}^\ell} t_{e,i}$, where the sum is over $\mathbb{F}_p$. Therefore, we can express $\langle c_{e,i} \rangle$ as a hypercube sum of $\langle t_{e,i} \rangle$ reduced modulo $p$, and thus, also as a hypercube-sum of $T_{e,i}(\langle e \rangle)$, modulo prime $p$.

Recall that $\text{FP} \subseteq \#P$ (Definition B.1). Now, invoke Lemma B.3(3) to obtain a $\#P/\text{poly}$ function that computes the hypercube-sum. Since modular arithmetic can be efficiently simulated by poly($\log p$) size Boolean circuits, residue of the hypercube-sum modulo $p$ can be computed easily by a function in $\#P/\text{poly}$ [GS11]. Composition of the two together yields the desired function $\phi_{g,i}(\langle e \rangle)$, and the closure property discussed in Lemma B.3(4) proves that it is in $\#P/\text{poly}$.
3.3 Explicitness over rationals: A weaker presentable border (Theorem 3)

In the introduction, we deliberated that extending the proof to rationals necessitates a restriction on coefficients of the polynomial as well as the computation model. To formalize the former, we define weight of an integer \( a \in \mathbb{Z} \) as \( \text{wt}(a) := 1 + \lceil \log_2 |a| \rceil \), to denote the number of bits required in its (signed) binary representation. We can naturally extend it to rationals \( t \in \mathbb{Q} \) as \( \text{wt}(t) := \min_{a,b \in \mathbb{Z}} \{ \text{wt}(a) + \text{wt}(b) : t = a/b \} \). Further, we define the weight of a polynomial \( g(y) \) as the maximum weight of all its rational coefficients, \( \text{wt}(g) := \max_e \{ \text{wt}(c_e) : c_e \text{ is the coefficient of } y^e \text{ in } g \} \).

As for the restriction on the model, we consider the following constrained ‘presentable’ border class, where the verifier circuit of the approximating polynomial has bounded weight and bitsize. Refer to Definition 1.2 for comparison.

**Definition 3.10 (Weak Presentable VNP).** The presentable border class \( \overline{\text{VNP}}_{\text{wk}} \), 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

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

for some error \( Q \in \mathbb{F}[\varepsilon][x_1, \ldots, x_n] \) and order \( M \in \mathbb{N} \) such that \( \text{wt}(g) \) is at most \( \text{poly}(n) \). Moreover, there exists a verifier polynomial \( h \in \mathbb{F}[x_1, \ldots, x_n, y_1, \ldots, y_m, \varepsilon] \) with \( m, \deg_{x,y}(h) \), and \( \text{bitsize}_\mathbb{F}(h) \) all bounded by \( \text{poly}(n) \) satisfying a hypercube-sum expression

\[
\sum_{\alpha \in \{0,1\}^m} h(x, \alpha, \varepsilon) = g(x, \varepsilon).
\]

We denote the tuple \((m, \text{bitsize}(h), \text{wt}(h))\) as size parameters for the \( \overline{\text{VNP}}_{\text{wk}} \) model. Since \( \text{size}(h) \) is at most \( \text{bitsize}(h) \), it is immediate that \( \overline{\text{VNP}}_{\text{wk}} \subseteq \overline{\text{VNP}}_\varepsilon \). It is called ‘weak’ because it bounds the weight of the computed polynomial \( f(x) \) by \( \text{poly}(n) \); otherwise, it could easily be exponential in \( n \).

The following theorem shows that polynomials in \( \overline{\text{VNP}}_{\text{wk}} \) over \( \mathbb{Q} \) are explicit.

**Theorem 3 (Explicit over \( \mathbb{Q} \)).** Let \( f \in \mathbb{Q}[x_1, \ldots, x_n] \) be a polynomial in \( \overline{\text{VNP}}_{\text{wk}} \). Then \( f \) is in \( VNP \).

In the rest of the section we give the proof of the theorem. Suppose \( g \in \mathbb{Q}[\varepsilon, x_1, \ldots, x_n] \) is the polynomial approximating \( f \) as per Definition 3.10. Let the size parameters be \((s, s, s)\), where \( s := \text{poly}(n) \), thus, \( \text{wt}(g) \leq s \). Choose a prime \( p \) such that \( 2s < \text{wt}(p) \leq 2s + 2 \). Existence of a prime in this range is easily guaranteed by Bertrand’s postulate (refer to [AZ18, Chapter 2]).
Recall that for an exponent vector $e \in \mathbb{N}^n$, we denote $c_e = a/b$, for at least $a, b \in \mathbb{Z}$, as the coefficient of $e^M x^e$ in $g$. Let $\phi_p(\langle e \rangle)$ be the reduced coefficient function defined as follows: $\phi_p(\langle e \rangle) \equiv \langle c_e \rangle \mod p$.

By the choice of the prime we know that $0 < b < p$, so, $b \not\equiv 0 \mod p$. Hence in the above equation, $\phi_p(\langle e \rangle)$ outputs the integral coefficient of $g$ reduced to $\mathbb{F}_p$. From Lemma 3.2, and the fact that $p$ is a $O(s)$–bit number, we get that $\phi_p$ belongs in $\#P/poly$. Moreover, the following lemma extends it further, so as to recover the coefficients uniquely over $\mathbb{Q}$ using a $\#P/poly$ function.

**Claim 3.11** (Rational coefficient function). There is a function $\phi_g$ in $\#P/\text{poly}$ which computes the encoding of a unique rational number $c_e = a/b$, $a, b \in \mathbb{Z}$, such that $\langle c_e \rangle \equiv \phi_p(\langle e \rangle) \mod p$ and $\text{wt}(a/b)$ is at most $s$.

**Proof.** For the uniqueness, let $a \neq \hat{a}$ and $b \neq \hat{b}$ be $s$–bit integers such that $a/b \equiv \hat{a}/\hat{b} \equiv \phi_p(\langle e \rangle) \mod p$. Comparing the ratios, we get $\hat{a}b - \hat{b}a \equiv 0 \mod p$. Recall that $p > 2^{2s}$, so, $\hat{a}b - \hat{b}a < p$, therefore, $a/b = \hat{a}/\hat{b}$ as absolute rationals.

Let $\psi_p$ be a verifier function that takes binary inputs $a$ and $b$, defined as follows: If $\gcd(a, b)$ is non-zero, then output 0. Otherwise, output the pair $(a, b)$ if $a \equiv b \cdot \phi_p(\langle e \rangle) \mod p$. Given the output of $\phi_p$, such a verification is possible in polynomial time. Since $\phi_p \in \#P/poly$, we get $\psi_p \in \#P/poly$. Using such a verifier function, define the required coefficient function $\phi_g$ as follows:

$$
\phi_g(\langle e \rangle) = \sum_{a, b \in \{0, 1\}^s} \psi_p(\langle e \rangle, a, b).
$$

We emphasize here that the coprimality condition in the definition of $\psi_p$, ensures a non-zero output for a unique pair of inputs in the hypercube-sum above (we do not overcount the same rational numbers again). Finally, from the discussion on closure properties of $\#P/\text{poly}$ in Lemma B.3(3), we can conclude that $\phi_g \in \#P/poly$. \hfill \Box

**Proof of Theorem 3.** From the preceding discussion we obtain a coefficient function $\phi_g$ for approximating polynomial $g$. As before, it is helpful to think of $\phi_g$ returning $a/b$ as a binary encoding of the tuple $(a, b)$ representing the rational coefficients. We observe, as in Equation 3.4, that if $f = \sum_e c_e \cdot x^e$, then $\langle c_e \rangle = \phi_g(\langle M, e_1, \ldots, e_n \rangle)$. Using Valiant’s criterion in Proposition 2.1 we conclude that $f$ is in VNP. \hfill \Box

**Number field extensions.** A number field $\mathbb{Q}(\alpha) = \mathbb{Q}[\alpha]$ is a finite degree extension of $\mathbb{Q}$, for some algebraic number $\alpha \in \mathbb{C}$. If $\alpha$ is a root of a monic irreducible polynomial $u$ over $\mathbb{Q}$ of degree $m$, then

$$
\mathbb{Q}[\alpha] = \{r_0 + r_1 \alpha + \ldots r_{m-1} \alpha^{m-1} : r_i \in \mathbb{Q}, \forall i \in [d] \},
$$

where the representation is unique due to the $\mathbb{Q}$-basis.
The weight of the elements here is defined in the same way as that of the rational polynomial $r_0 + r_1 \alpha + \ldots r_{m-1} \alpha^{m-1}$. So, we can talk about the $\overline{\text{VNP}}_{\text{wk}}$ model over number fields as well.

We can prove the following theorem analogous to Theorem 3 over number fields. Note that our theorem is unconditional for ‘small’ number fields like $\mathbb{Q}(\sqrt{-1})$, $\mathbb{Q}(\sqrt{2})$, $\mathbb{Q}(\sqrt{3})$, $\mathbb{Q}(\zeta_{13})$, etc.

**Theorem 4** (Explicit over number fields). Let $f \in \mathbb{Q}[\alpha][x_1, \ldots, x_n]$ be a polynomial in $\overline{\text{VNP}}_{\text{wk}}$. Assuming Generalized Riemann Hypothesis (GRH), $f$ is in $\text{VNP}$.

**Proof Sketch.** We will only sketch the proof, as it follows the same structure as before. Suppose $g \in \mathbb{Q}[\alpha][x_1, \ldots, x_n]$ is a polynomial approximating $f$ as per Definition 3.10. As earlier, we chose a prime $p$ in the range $2s < \text{wt}(p) \leq O(s)$, so that $u_0 \equiv u \mod p$ remains irreducible, and of the same degree $m$. The existence of such a prime is guaranteed by the famous Chebotarev’s Density Theorem, with an explicit version assuming Generalized Riemann Hypothesis (GRH), refer to [LO77, Theorem 1.1]. Moreover, the GRH assumption is not required if the degree $m$ is constant, see [LO77, Theorem 1.4].

Once again we can associate all the coefficients of $g$ as $\phi_p(e) \equiv c_e \mod u_0$. Basically, $\phi_p(e)$ outputs the coefficient of $g$ over $\mathbb{F}_q$, where $q := p^m$. So, we have reduced to the finite field case again. From Lemma 3.2 and weight upper bound on $p$, we get that $\phi_p$ is in $\#P/poly$.

Almost the same argument as in Claim 3.11 will prove that there exists a function $\phi_g$ in $\#P/poly$ that uniquely recovers the number field coefficients. Finally, we can use Valiant’s criterion (Proposition 2.1) to finish the proof.

**3.4 An application to deborder factors: Proof of Corollary 1.3**

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

**Definition 3.12 (Presentable $\overline{\text{VP}}$).** The presentable border class $\overline{\text{VP}}_\varepsilon$ 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

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

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

Although, $g$ has a small size circuit, we emphasise that the degree of $\varepsilon$-polynomials in $g$ is unrestricted. Further, it is apparent from the definitions that $\text{VP} \subseteq \overline{\text{VP}}_\varepsilon \subseteq \text{VNP}_\varepsilon$. Bürgisser in [Bü04, Theorem 1.3] proved that the class $\overline{\text{VP}}_\varepsilon$ contains all the low-degree separable factors of circuits of small size.

**Lemma 3.13.** 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 size$(f)$

---

Footnote: i.e. factor $u$ which is irreducible and has multiplicity coprime to the characteristic $p$. This isn’t an issue in characteristic zero fields.
and \( \deg(u) \) is at most \( s := \text{poly}(n, \log q) \). Then we have \( u \) in \( \overline{\text{VP}}_\varepsilon \).

**Remark.** We make a few observations.

1. In case \( f = u^e \), Kaltofen [Kal87] showed that \( u \) is \( \text{VP} \).

2. Bürgisser [Bü04] proved that \( u \) (in the lemma above) is in \( \overline{\text{VP}} \). 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 \( u \in \overline{\text{VP}}_\varepsilon \) in the appendix.

3. Over rationals, additionally assuming that \( \text{bitsize}(f) + \text{wt}(f) \) is bounded by \( \text{poly}(n) \) proves that \( u \) is in \( \overline{\text{VP}}_{\text{wk}} \), with no need to put conditions on \( e \) anymore. The problem to prove explicitness over rationals without the weight restriction remains open.

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 \( \text{VNP} \).

Over rationals, consider \( f \in \mathbb{Q}[x_1, \ldots, x_n] \) and its factor \( u \) such that \( \text{bitsize}(f), \text{wt}(f), \deg(u) \leq \text{poly}(n) \). Then, \( u \) is in \( \text{VNP} \).

**Proof.** We learn from Lemma 3.13 that the polynomial family \( u \in \overline{\text{VP}}_\varepsilon \). Moreover, \( \overline{\text{VP}}_\varepsilon \) is contained in \( \overline{\text{VNP}}_\varepsilon \) by definition. As over \( \mathbb{F}_q \), Theorem 1 proves \( \overline{\text{VNP}}_\varepsilon = \text{VNP} \), hence \( u \in \text{VNP} \).

Over rationals, once again we use Lemma 3.13 (Remark 3) to learn that the polynomial family \( u \in \overline{\text{VP}}_{\text{wk}} \). The class \( \overline{\text{VP}}_{\text{wk}} \) is contained in \( \overline{\text{VNP}}_{\text{wk}} \) and using Theorem 3 we prove that \( u \in \text{VNP} \). \( \square \)

4 **VNP is factor closed: Proof of Theorem 2**

In a pioneering work, Valiant [Val79], defined \( \text{VNP} \) as a class of polynomials which can be expressed as hypercube sum of a \( \text{VP} \) circuit (Definition 1.1). In a subsequent work, [Val82] showed that \( \text{VNP} \) 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 4.1 (VNP closure properties).** For all \( i \in [t] \), let \( f_i \in \mathbb{F}[x_1, \ldots, x_n, y_1, \ldots, y_m] \) be polynomials in \( \text{VNP} \) 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 \( \text{VNP} \).
2. Coefficient Extraction: For all \( i \in [t] \), let \( f_i = \sum_e c_e(x) \cdot y^e \). Then for all exponent vectors \( e \), the coefficient \( c_e \) 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.

The proof of the lemma is given in Appendix A. Meanwhile 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 4.2** (Prime power). Let \( f \in \mathbb{F}_q[x_1, \ldots, x_n] \) be a polynomial in VNP. If there is a polynomial \( u \) and an arbitrary positive integer \( e \) such that \( f = u^e \), then the factor \( u \) is in VNP.

**Lemma 4.3** (Coprime factors). Let \( f \in \mathbb{F}_q[x_1, \ldots, x_n] \) be a polynomial in VNP. If there are co-prime polynomials \( u \) and \( v \) such that \( f = u \cdot v \), then the factor \( u \) is in VNP.

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

**Lemma 4.4** (Any power). Let \( f \in \mathbb{F}_q[x_1, \ldots, x_n] \) be a polynomial in VNP. If there is a polynomial \( u \) and an arbitrary positive integer \( e \) such that \( f = u^e \), then the factor \( u \) is in VNP.

**Proof.** Let \( e := p^j \cdot \tilde{e} \), and \( u_1 := u^{\tilde{e}} \), such that \( p \) does not divide \( \tilde{e} \). Note that, when \( \tilde{e} = 1 \) then **Lemma 4.2** finishes the proof. When \( \tilde{e} > 1 \), we associate a polynomial \( \tilde{f} \) with a new variable \( z \) as follows:

\[
\tilde{f} := z^{\tilde{e}} - f = z^{\tilde{e}} - u_1^{\tilde{e}} \\
= (z - u_1) \cdot \left( z^{\tilde{e}-1} + z^{\tilde{e}-2}u_1 + \cdots + u_1^{\tilde{e}-1} \right) \\
= : u_2(z) \cdot u_3(z).
\]

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 \( u_3(u_1) = \tilde{e} \cdot u_1^{\tilde{e}-1} = 0 \). However, since \( \tilde{e} > 1 \) and \( u_1 \) is non-zero, it follows that the characteristic \( p \) divides \( \tilde{e} \), which contradicts our choice of \( \tilde{e} \).

Observe that \( z^{\tilde{e}} \) is trivially in VNP, hence we obtain that \( \tilde{f} \) is in VNP. Since \( u_2 \) and \( u_3 \) are co-prime, we invoke **Lemma 4.3** to shows that \( u_2 \) is in VNP, and therefore \( u_1 \) is in VNP. We finish the proof by using **Lemma 4.2** on \( u_1 \) to finally prove that \( u \) is in VNP.

With all the essential ingredients in place, we are now ready to prove the second main result of our paper. We will restate Theorem 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 4.4 proves that \( u \) is in VNP.

If \( u \) and \( v \) are coprime, then we conclude the proof using Lemma 4.3 and Lemma 4.4.

In the last case, there exists an irreducible polynomial \( w \in \mathbb{F}_q[x_1, \ldots, x_n] \) that divides both \( u \) and \( v \). Consider \( u_1 := w^e' \) and \( v_1 := (f/u_1) \) such that \( u_1, v_1 \) are coprime factors of \( f \). Again, using Lemma 4.3 and Lemma 4.4 we get that \( w \) is in VNP. Repeat this for all the irreducible factors of \( u \), and use the fact that VNP is closed under multiplication (Lemma 4.1); this concludes the proof of \( u \) being in VNP.

\[ \square \]

### 4.1 Factoring prime powers or Valiant’s converse: Proof of Lemma 4.2

To prove Lemma 4.2, we show that the coefficients of the factor polynomial \( u \) can be computed effectively, and thus use Valiant’s criterion to prove the claim. We will argue that coefficients of \( u \) can be obtained from the coefficient function of \( f \). Therefore, it would suffice to design an effectively computable coefficient function for \( f \), give that it is in VNP. To that effect, we prove the converse of Valiant’s criterion, over finite fields.

**Lemma 4.5 (Converse of Valiant’s criterion).** Let \( f = \sum_e c_e \cdot x^e \) be a polynomial in VNP over \( \mathbb{F}_q \). Then, there exists a function \( \phi_f \) in \( \#P/poly \) such that for all \( e \), \( \phi_f(\langle e \rangle) = \langle c_e \rangle \).

**Proof.** Let \( D := \deg(f) \) and the VNP size parameters of \( f \) be \((s, s)\) where \( s := \text{poly}(n, \log q) \). Using the exponential-interpolation in Lemma 3.1, with \( D = \text{poly}(s) \), we can prove that each coefficient \( c_e \) of \( f \) is a hypercube-sum of small-circuit evaluations, with parameters \((\text{poly}(s), \text{poly}(s))\)\(^8\). That is, there is a polynomial \( t_e \) over a finite field extension \( \mathbb{F}_{q'} \), \( q' \leq \text{poly}(s) \), such that

\[
c_e = \sum_{b \in \{0,1\}^\ell} t_e(b_1, \ldots, b_\ell),
\]

where \( \ell \) and size\((t_e)\) are at most \( \text{poly}(s) \). Next, moving to the boolean world, Lemma 3.2 shows that such an algebraic representation can be transformed to obtain the coefficient function \( \phi_f \in \#P/poly \) such that \( \phi_f(\langle e \rangle) = \langle c_e \rangle \).

\[ \square \]

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

---

\(^8\)The same conclusion can be made from VNP closure properties stated in Lemma 4.1.

21
We now have all the necessary tools needed to prove the lemma.

**Proof of Lemma 4.2.** Given that \( f = u^{p^j} \), let \( u =: \sum_{a \in L} c_a x^a \), where the support \( L \) represents the set of exponent vectors associated to \( u \). Essentially, Claim 4.6 allows us to distribute the prime power over addition as follows:

\[
  f = u^{p^j} = \left( \sum_{a \in L} c_a \cdot x^a \right)^{p^j} = \sum_{a \in L} (c_a)^{p^j} x^{p^j \cdot a}.
\]

The last expression above clearly associates the coefficients of \( x^{p^j \cdot a} \) in \( f \) to coefficients of \( x^a \) in \( u \). Since \( f \) is in \( \text{VNP} \), Lemma 4.5 guarantees a \( \#P/\text{poly} \) function \( \phi_f \) such that the following congruence, in the finite field \( \mathbb{F}_q \), is true for all \( a \in L \):

\[
  (\phi_f (p^j \cdot a))^{1/p^j} = \phi_f(p^j \cdot a)^{q/p^j} = \phi_f(p^j \cdot a)^{p^j - 1} =: \phi_u(a) = \langle c_a \rangle.
\]

In Lemma B.3 it was proved that \( \#P/\text{poly} \) functions are closed under repeated-squaring, hence we conclude that \( \phi_u \in \#P/\text{poly} \). Invoking Proposition 2.1 on \( \phi_u \) proves that the factor \( u \in \text{VNP} \).

4.2 Factoring co-prime factors: Proof of Lemma 4.3

The proof of Lemma 4.3 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 \( \text{VNP} \) 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 \( \tau_\alpha : \mathbb{F}_q[x_1, \ldots, x_n] \rightarrow \mathbb{F}_q[x, x_1, \ldots, x_n] \) 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 \( c_e \) of the leading term \( x^d \) of \( f_\alpha \) gives

\[
  c_e =: \sum_{|e|=d} \tilde{c}_e \cdot \alpha_1^{e_1} \cdots \alpha_n^{e_n}.
\]

PIT lemma guarantees that with high probability, a random choice of \( \alpha \) ensures \( c_e \) is a non-zero field element (refer to [SY10, Lemma 4.2]). Then, \( f_\alpha/c_e \) is a monic polynomial in \( x \). Further, if \( (s, s) \) is the \( \text{VNP} \) size parameters of \( f \), then the parameters for \( f_\alpha \) are \( (s, s + O(n)) \). When the field is too small, to pick the right \( \alpha \), we can obtain it from a field extension \( K \) of degree at most \( \text{poly}(\deg(f)) \). Since arithmetic operations over \( K \) can be efficiently simulated in \( \mathbb{F} \) (refer to [Bü00, Proposition 4.1]), we will assume \( K = \mathbb{F}_q \) 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 \( f_\alpha/c_e \) as \( f_\alpha \) and associate a polynomial
$\tilde{f} \in \mathbb{F}_q[x_1, \ldots, x_n][x, y]$ as follows: $\tilde{f}(x, y) := f_\alpha(x, yx_1 + a_1, yx_2 + a_2, \ldots, yx_n + a_n)$, where $a \in \mathbb{F}_q^n$ is a point.

If $f_\alpha$ is monic and $u_\alpha$ 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 $\tilde{f}$, 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 4.7** (Initialize Hensel lifting). Let $f = u \cdot v$ be such that $u, v$ are co-prime polynomials. Then the associated univariate factors $\bar{u}(x, 0)$ and $\bar{v}(x, 0)$ of $\tilde{f}(x, 0)$ 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 $\tilde{f}(x, y)$ remains monic in $x$ and is in VNP with size parameters $(s, s + O(n))$.

**Hensel's Lifting.** Let us re-assign $f = \tilde{f}$ for notational simplicity. Recall that $f(x, y)$ is monic in $x$, therefore $f_0 := f(x, 0) \in \mathbb{F}_q[x]$ is a univariate polynomial of degree $d$. Since $f_0$ can have at most $d$ factors, $u_0 := u(x, 0)$ and $v_0 := v(x, 0)$ are in VNP with parameters $(1, O(d))$. We will use the following ever-famous Hensel’s Lifting lemma from number theory to lift the roots uniquely (mod $y$). For a detailed discussion on the specific monic version of the Lifting lemma required for our proof, we encourage the readers to refer [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 4.8** (Monic Hensel's Lifting). Let $f = u \cdot v \in \mathbb{K}[x, y]$ be such that $u, v$ are co-prime, and $u$ is monic in $x$. Additionally, we are given $u_0 \equiv u \mod y$ and $v_0 \equiv v \mod y$ such that $a_0u_0 + b_0v_0 \equiv 1 \mod y$. 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. $u_k \equiv u_{k-1} \mod y^{2^k-1}$ and $v_k \equiv v_{k-1} \mod y^{2^k-1}$.
2. $f \equiv u_k \cdot v_k \mod y^{2^k}$ such that $a_ku_k + b_kv_k \equiv 1 \mod y^{2^k}$ and $u_k$ is monic in $x$.
3. $u_k \equiv u \mod y^{2^k}$ and $v_k \equiv v \mod y^{2^k}$.

Moreover, for every $k$, the lifted factors $u_k$ and $v_k$ are unique polynomials mod $y^{2^k}$.

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

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

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

\[ u_k := u_{k-1} + y^{2^{k-1}} r_k \]  \hspace{1cm} (4.9)
\[ v_k := \tilde{v}_k \left( 1 + y^{2^{k-1}} q_k \right). \] \hspace{1cm} (4.10)

It is easy to verify that they are the valid lifts as per Lemma 4.8. 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 \( f \in \text{VNP} \), the factor \( u \in \text{VNP} \) 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 4.11** (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 \( f =: \sum_{e_i \in L} c_{e_i}(x_1, \ldots, x_n) \cdot x^{e_1} y^{e_2} \).

Given the coefficients \( c_{e_1}, \ldots, c_{e_{|L|}} \) as input, there exists a circuit \( C_{u_t}^{(t)} \) over \( \mathbb{F}_q \) which computes \( \text{Hom}_{\leq d}(u_t)^9 \). Further, there is a constant \( \beta \geq 2 \) such that the size of the circuit \( C_{u_t}^{(t)} \) is at most \( \text{poly}(d, \beta^t) \), 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 \( s_f := \text{poly}(d) \) that computes \( f \). Then, the proof is an easy consequence of the following inductive analysis on \( t \).

The base case is easy to analyse. Let \( C_{u_t}^{(t-1)}, C_{v_t}^{(t-1)}, C_a^{(t-1)} \), 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 4.8. 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 \( s_f + O(s_{t-1}) \) \(^9\). Then observe that size(\( \bar{u}_t \)) and 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 4.9 and 4.10. 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 \( C_{a_t}^{(t)} \) and \( C_{b_t}^{(t)} \) computing \( a_t \) and \( b_t \).

\(^9\)This is the sum of the homogeneous parts of \( u_t \) up to degree \( d \).
\(^10\)For notations, refer to the discussion proceeding Lemma 4.8.
Lastly, we homogenize $C_t^u$ using Lemma A.1, 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 $\beta t$ 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 4.3.

**Lemma 4.3 (restated).** Let $f \in \mathbb{F}_q[x_1, \ldots, x_n]$ be a polynomial in VNP. If there are co-prime polynomials $u$ and $v$ such that $f = u \cdot v$, then the factor $u$ is in VNP.

**Proof of Lemma 4.3.** Assume that $f \in \mathbb{K}[x, y]$ after all the necessary invertible transformations discussed earlier in the section to apply Lemma 4.8. Let support $L$ be the set of exponent vectors of $f$ such that $f =: \sum_{e_i \in L} c_{e_i}(x_1, \ldots, x_n) \cdot x^{e_1} y^{e_2}$.

Using Lemma 4.11 with $t \geq \log(\deg(f)) + 1$ gives a circuit $C_t^u$ 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 $f \in \text{VNP}$, Lemma 4.1(2) shows that the coefficients $c_{e_i} \in \text{VNP}$. Moreover, Lemma 4.1(3) will prove that $C_t^u$ composed with VNP polynomials, remains in VNP. Therefore, the factor $u$ is in VNP.

**5 Conclusion**

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

As an application of our debordering result, we advance partially towards proving the factor conjecture [Bü00, Conjecture 8.3], by showing that low-degree ‘non-singular’ factors of small size circuits are explicit. This still does not rule out the possibility: Could the Permanent polynomial be a factor of a small circuit of exponential degree?

Our debordering technique, of exponential interpolation, further proved that over all characteristics, VNP is closed under factoring, and thus resolves Bürgisser’s conjecture [Bü00, Conjecture 2.1].

**Whitebox PIT.** Our newly introduced presentable border classes open up a new avenue of studying Polynomial Identity Testing (PIT) in the whitebox setting. PIT is a fundamental problem in complexity theory, that decides the zeroness of the given polynomial (refer [Sax09, Sax14] and also [SY10, Chapter 4]). It is studied under two well known settings: Blackbox and Whitebox. The former allows only evaluations, while the latter allows to look at the inner structure of the model. PIT on border classes, naturally extends to testing the zeroness of a polynomial, given
its approximating polynomial. Concretely, let \( g \) approximate a non-zero polynomial \( f \in \overline{\text{VP}} \), then there exists an evaluation point \( \alpha \) such that \( g(\alpha, \varepsilon) \) is not a multiple of \( \varepsilon \). We emphasise that mere non-zeroness of \( g(\alpha, \varepsilon) \) does not guarantee non-zeroness of \( f \). For a comprehensive discussion and motivations of blackbox border PIT, refer [FS18, DDS21].

The arbitrarily large complexity of \( \varepsilon \)-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 \( \overline{\text{VP}} \varepsilon \) constrain the \( \varepsilon \)-polynomials, and therefore we make the whitebox setting interesting. It is worthwhile to investigate whitebox PIT on presentable border classes; for instance, study constant depth circuits to begin with.

References


A Structural results

Throughout the paper, we refer 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 \(\text{Hom}_k(f)\). Similarly, we define \(\text{Hom}_{\leq k}(f)\) equal to \(\sum_{i \in [k]} \text{Hom}_k(f)\). The following well-known structural result proves that given a blackbox access to a circuit computing the polynomial \(f\), we can construct a circuit that computes all its homogeneous components. Refer [SY10, Theorem 2.2] for the proof.

**Lemma A.1 (Homogenisation).** Consider an \(n\)-variate polynomial \(f := \sum_{i \in [d]} c_i(y)x_i\) computable by a circuit of size \(s\) over \(\mathbb{F}\). Then \(\text{size}(c_i)\) is at most \(\text{poly}(s, n, d)\), for all \(i \in [d]\). Moreover, \(\text{size}(\text{Hom}_{\leq d}(f))\) is at most \(\text{poly}(s, n, d)\).

We will invoke the lemma to homogenise the Hensel lifting circuit, constructed in the proof of Lemma 4.11. A straight-forward application of the homogenisation lemma is the elimination of division gates and computing derivatives. Refer [SY10, Theorem 2.11].

**Lemma A.2 (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 \(f \mod (x)^{d+1}\) can be computed by a circuit of size \(\text{poly}(s, d)\).

**Lemma A.3 (Derivatives).** Consider an \(n\)-variate polynomial \(f \in \mathbb{F}[y, x_1, \ldots, x_n]\) computable by a circuit of size \(s\) over \(\mathbb{F}\). Then for any \(k\), \(\partial_y^k f\) can be computed by a circuit of size \(\text{poly}(s, k)\).

Hypercube-sum of Formulas. An algebraic circuit is called a formula, if the underlying graph is a tree. In Section 1 we defined the class \(\text{VNP}\) as hypercube-sum of small size circuits. Valiant proved in [Val82] that these polynomials can be equivalently computed by a hypercube-sum of small size formulas. Refer [Bü00, Theorem 2.13] and [MP08, Theorem 2] for the proof. A direct consequence of the equivalence is the following structural lemma, that helps in proving closure properties of \(\text{VNP}\).

**Lemma A.4 (Verifier formula).** Consider an \(n\)-variate polynomial \(f\) of degree \(d\) computable by a circuit of size \(s\) over \(\mathbb{F}\). Then, there is a verifier polynomial \(h\), with \(m\) and the formula size of \(h\) both bounded by \(\text{poly}(s, n, d)\), satisfying the hypercube-sum expression

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

Closure properties of \(\text{VNP}\). In Section 4 we discussed various closure properties of polynomials in the class \(\text{VNP}\). We invoke these closure properties to proof Theorem 2. The Composition property is particularly crucial in the proof of Lemma 4.3.

**Proof of Lemma 4.1.** The statements can be proved directly from the Definition of \(\text{VNP}\).
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 $\poly(n, m)$. Then the properties can be proved as follows.

**Addition and Multiplication:** Observe that 

$$f_+ = \sum_{i \in [t]} f_i = \sum_{i \in [t]} \left( \sum_{a_i \in \{0, 1\}^{m_i}} h_i(x, a_i) \right) = \sum_{(a_1, \ldots, a_t) \in \{0, 1\}^{\ell_+}} h_+(x, a_1, \ldots, a_t),$$

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

$$f_\times = \prod_{i \in [t]} f_i = \prod_{i \in [t]} \left( \sum_{a_i \in \{0, 1\}^{m_i}} h_i(x, a_i) \right) = \sum_{(a_1, \ldots, a_t) \in \{0, 1\}^{\ell_\times}} h_\times(x, a_1, \ldots, a_t).$$

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

**Coefficient Extraction:** The proof runs the same as the proof of Lemma 3.1, with both $s$ and $D$ at most $\poly(n, m)$. 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 $v \in \text{VP}$. Invoke Lemma A.4 on the circuit $C$ for the verifier polynomial $v$ to obtain a polynomial $h$ and $\ell \leq \poly(t, d)$ satisfying 

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

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

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

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.

$$C(f_1, \ldots, f_t) = \sum_{a, a_i \in \{0, 1\}^{\ell'}} T(h_1(x, a_1), \ldots, h_\ell(x, a_\ell), a),$$

where $\ell' = \ell + \sum_i m_i \leq \poly(t, d, n, m)$. Moreover, the size of the circuit computing $T$ composed
with \( h_1, \ldots, h_t \) is at most \( O(\text{size}(T) + \sum_i \text{size}(h_i)) \leq \text{poly}(t, d, n, m) \). 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 \(^{11}\).

\[ \square \]

\section*{B 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 \( r \).

\textbf{Definition B.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 \( \chi \in \mathcal{P} \) satisfying \( \psi(x) = \langle |S| \rangle \) where

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

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

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.

\textbf{Definition B.2} (Non-uniform complexity classes). The complexity class \( \mathcal{C}/\text{poly} \) is defined as the set of functions \( \phi: \{0,1\}^* \to \{0,1\}^* \) such that there exists a \( \psi \) in class \( \mathcal{C} \) and a polynomial length advice function \( \alpha: \mathbb{N} \to \{0,1\}^* \) satisfying \( \phi(x) = \psi(x, \alpha(|x|)) \).

We remark that the advice function \( \alpha \) in the definition above only depends on the length of the input. Moreover, for all \( n \in \mathbb{N} \), \( |\alpha(n)| \leq \text{poly}(n) \). The following lemma shows that the complexity classes of our interest are closed under usual operations.

\textbf{Lemma B.3} (Closure Properties). For a positive integer \( r \), consider a set of functions \( \phi_1, \ldots, \phi_r \) in \( \#P/\text{poly} \). Consider an input string \( x \in \{0,1\}^* \). Then the following closure properties can be shown:

1. Addition and Multiplication: Let \( \phi_+ (x) := \sum_{i \in [r]} \phi_i(x) \) and \( \phi \times := \prod_{i \in [r]} \phi_i(x) \). Then, \( \phi_+ \) and \( \phi \times \) are also in \( \#P/\text{poly} \) for \( r \leq \text{poly}(|x|) \).

\(^{11}\)Consider a pedagogical example, \( C(z) = z^2 \) from [CKS19b, Footnote 9].
2. Repeated Squaring: For all $i \in [r]$, $\phi_i(x)^t$ is in $\#P/poly$ for $t \leq 2^{\text{poly}(|x|)}$.

3. Projection: Let $\Phi_i(x) := \sum_{b \in \{0,1\}} \phi_i(x, b)$, where $\ell \leq \text{poly}(|x|)$. Then $\Phi_i$ is in $\#P/poly$.

4. Composition: For all $i, j \in [r]$, $\phi_i(\phi_j(x))$ is in $\#P/poly$

Proof. For every $\#P/poly$ function $\phi_i$, let $\alpha_i$ be its advice function and $\chi_i$ be its associated language in $P$ defining the counting set $S_i$, see Definition B.1.

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

\[
S_+ := \{(i, y) \in \{0,1\}^{\text{poly}(|x|)+\log r} : (x, \alpha(|x|, i), y) \in \chi_i\}, \quad \text{and} \quad \nu \in [r], (x, \alpha(|x|, \langle i \rangle), y_i) \in \chi_i \}.
\]

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

\[
\begin{align*}
\phi_+(x) &= \psi_+(x, \bar{\alpha}(|x|)) := \langle |S_+| \rangle, \quad \text{and} \\
\phi_\times(x) &= \psi_\times(x, \bar{\alpha}(|x|)) := \langle |S_\times| \rangle.
\end{align*}
\]

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

2. Repeated Squaring: Note that $\phi_i(x)^2$ is in $\#P/poly$ from the discussion on multiplication above. Then the claim follows by repeatedly multiplying $\#P/poly$ 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 $x$, it will be same throughout the hypercube-sum. This essentially, lets us add exponentially many $\#P/poly$ function. Let $\Psi_i(x, \alpha_i(|x|)) = \langle |S_p| \rangle$ where

\[
S_p := \{(b, y) \in \{0,1\}^{\text{poly}(|x|)+\ell} : (x, \alpha_i(|x|), b, y) \in \chi_i\}.
\]

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

4. Composition: Let $\bar{\alpha}(|x|) = (\alpha_i(|x|), \alpha_j(|x|))$ be the advice function. Define $\Psi(x, \bar{\alpha}(|x|)) = \langle |S_c| \rangle$ where

\[
S_c := \{(y, z) \in \{0,1\}^{\text{poly}(|x|)} : (x, \alpha_i(|x|), y) \in \chi_j \text{ and } (\phi_j(x), z) \in \chi_i\}.
\]

Clearly, $\phi_i(\phi_j(x)) = \Psi(x, \bar{\alpha}(|x|))$. Moreover, given that $\chi_i, \chi_j \in \mathbb{P}$, the conjugation in the set $S_c$ can be verified in $\mathbb{P}$. Hence, the composed functions are in $\#P/poly$. 

36
In the proof of Lemma 3.2, we claimed that string function in $\#P/poly$ are closed under concatenation. In the following claim we formalise it. For binary string $a, b \in \{0,1\}^*$, we use $\langle a, b \rangle$ to denote string concatenation.

**Claim B.4.** For a positive integer $r$, consider a set of functions $\phi_i : \{0,1\}^* \to \{0,1\}^{r \ell}$ in $\#P/poly$. Define a map $\Phi : \{0,1\}^* \to \{0,1\}^{r \ell}$ such that $\Phi(x) = \langle \phi_1(x), \ldots, \phi_r(x) \rangle$, for all $x \in \{0,1\}^*$. Then, $\Phi$ is in $\#P/poly$ for $r, \ell \leq poly(|x|)$.

**Proof.** There is a trivial function $\Psi$ in $FP$ that takes output of $\phi_i(x)$ as input and outputs $\langle \phi_1(x), \ldots, \phi_r(x) \rangle$ as a concatenated string. Clearly, $\Phi = \Psi(\phi_1(x), \ldots, \phi_r(x))$. Recall that $FP \subseteq \#P$, then Lemma B.3(4) proves that the composition of the functions is in $\#P/poly$. □

## C Low degree factors are easy to approximate

In this section we will sketch the proof of Lemma 3.13. 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 $\alpha_i, \beta_i \in \mathbb{F}_q$ 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 $k \in \mathbb{F}_q^\times, \gamma_i > 0$ 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, $\mu_i := h_i(\overline{0})$ 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 C.1.** 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 $0 \leq c_i \leq \gamma_i, k' \in \mathbb{F}_q^\times$, and $h_i \in \mathbb{F}_q[[x_1, \ldots, x_n]]$.

Recall the definition of $\text{Hom}_{\leq d_u}(h_i)$ from Appendix A. Observe that

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

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 C.2.** For all \( i \in [d_f] \), there is an approximating polynomial \( g_i \in \mathbb{F}_q[\varepsilon][x_1, \ldots, x_n] \) satisfying \( g_i = \varepsilon^M \text{Hom}_{\leq d_u}(h_i) + \varepsilon^{M+1}Q_i(x, \varepsilon) \), for some error \( Q_i \in \mathbb{F}[\varepsilon][x_1, \ldots, x_n] \) and order \( M \in \mathbb{N} \). Moreover \( \deg_x(g_i) \leq d_u \) and \( \text{size}_x(g_i) \leq \text{poly}(d_u, \text{size}(f)) \).

**Proof Sketch.** Let \( \tilde{f} := \tau(f) \in \mathbb{F}_q[\varepsilon, y] \) and \( \mu_i := h_i(\overline{0}) \). Define a perturbed polynomial \( F := \tilde{f}(x, y + \mu_i + \varepsilon) - \tilde{f}(0, \mu_i + \varepsilon) \) over the ring \( \mathbb{F}_q[\varepsilon] \). Since \( \varepsilon \) 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:

\[
y_0 = 0, \quad y_{t+1} := y_t - \frac{F(x, y_t)}{\partial_y F(x, y_t)} \quad \text{(C.3)}
\]

where \( y_t \equiv H_j \mod \langle x \rangle^{2t} \). The quadratic convergence of degree in Newton iteration implies that it suffices to assume \( t \leq \log d_u + 1 \). 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 \( \text{Hom}_{\leq d_u}(H_j)|_{\varepsilon=0} = \text{Hom}_{\leq d_u}(h_i) \).

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 \( \tilde{g}_i \in R[x] \) computes \( A_t/B_t \equiv y_t \mod \langle x \rangle^{2t} \) and satisfies the following:

\[
\tilde{g}_i = \text{Hom}_{\leq d_u}(h_i) + \varepsilon \tilde{Q}_i(\varepsilon, x, y),
\]

for some error \( \tilde{Q}_i \in R[x_1, \ldots, x_n] \). 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 \( g_i = \varepsilon^M \text{Hom}_{\leq d_u}(h_i) + \varepsilon^{M+1}Q_i(\varepsilon, x, y) \). Moreover, \( g_i = \varepsilon^M \tilde{g}_i \). Therefore, the proposition follows easily by proving \( \text{size}_x(g_i) \leq \text{poly}(d_u, \text{size}(\tilde{f})) \), and \( M \leq 2\text{poly}(d_u) \). In case, \( \deg_x(g_i) \) 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 C.3. Treating \( \varepsilon \) as a variable, observe that \( \text{size}_\varepsilon(F) \leq s_0 := \text{size}(\tilde{f}) + 2 \). Homogenise the circuit computing \( F \) using Lemma A.1 with respect to \( y \) to obtain \( \text{Hom}_{\leq d_u}(F) \) of size \( \text{poly}(d_u, s_0) \). Use this homogenised circuit to obtain the circuit computing \( \partial_y F \) of size \( \text{poly}(d_u, s_0) \). Using division and subtraction gates, compute \( A_1/B_1 \) and let its size be \( s_1 := \max(\text{size}_\varepsilon(A_1), \text{size}_\varepsilon(B_1)) \leq \text{poly}(d_u, s_0) \). Let \( t = \log d_u + 1 \), then Newton iteration gives an easy recurrence on the size \( s_t \leq c + s_{t-1} + \text{poly}(d_u, s_0) \), where \( c \) is a small constant. Solving the recurrence gives \( s_t \leq \text{poly}(d_u, s_0) \leq \text{poly}(d_u, \text{size}(\tilde{f})) \). Finally, eliminate the division with respect to \( x, y \) variables using Lemma A.2 to obtain the circuit computing \( \tilde{g}_i \) of size at most \( \text{poly}(d_u, \text{size}(\tilde{f})) \).

The upper bound is preserved after the inverse transformation \( \tau^{-1} \).

**Order analysis.** Let the \( \varepsilon \) degree in \( A_1/B_1 \) be denoted by \( d_1 := \max(\deg_x(A_1), \deg_x(B_1)) \). 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 \( d_t \leq d_u \cdot d_{t-1} \), solving which gives \( d_t \leq (d_u)^{\log d_u} \leq \ldots \).
2^{\text{poly}(d_u)}. \text{ Then to obtain } g_i, \text{ set } M = d_t \leq 2^{\text{poly}(d_u)}.

We are now ready to prove Lemma 3.13. For two arbitrary polynomials \( u \) and \( h \), let \( \text{size}(u \mid h) \) denote the size of the circuit that computes \( u \) given \( h \) for free. The definition can be extended to \( \text{size}(u \mid h) \) naturally.

**Lemma 3.13** (restated). Let \( q := p^a \) and \( e \) be a positive integer coprime to \( p \). Consider a polynomial \( f \in \mathbb{F}_q[x_1, \ldots, x_n] \) 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 \( u \in \overline{\text{VP}}_\varepsilon \).

**Proof.** Using the map \( \tau \) defined earlier, Lemma C.1 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

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

then observe that \( \tau(u) \equiv H \mod (x)^{d_u+1} \). 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

\[
\overline{\text{size}}(H) \leq \overline{\text{size}}(H \mid \text{Hom}_{\leq d_u}(h_i)) + \overline{\text{size}}(\text{Hom}_{\leq d_u}(h_i)),
\]

see [Bü04, Lemma 2.3(3)]. Since \( d_u \leq s \), Proposition C.2 proves that \( \overline{\text{size}}(\text{Hom}_{\leq d_u}(h_i)) \) is at most \( \text{poly}(s) \). Then, clearly \( \overline{\text{size}}(H) \leq \text{poly}(s) \). Suppose \( G \) approximates \( H \), in the usual sense. Eliminate division in \( G \mod (x)^{d_u+1} \) using Lemma A.2 to obtain the approximation of \( \tau(u) \), moreover almost immediately we get that \( \overline{\text{size}}(\tau(u)) \leq \text{poly}(s, d_u) \). Since shifting and scaling do not change the size complexity, apply the inverse transformation to conclude that \( u \in \overline{\text{VP}}_\varepsilon \). \qed

**Extension.** In Lemma 3.13 we remarked that over rationals, low-degree factor \( u \) belongs to a weaker presentable class \( \overline{\text{VP}}_{\text{wk}} \) (refer Definition 3.10), provided \( \text{bitsize}(f) + \text{wt}(f) \) is bounded by \( \text{poly}(n) \). The proof is essentially the same as the finite fields case discussed above. The blow-up of field constants are controlled due to homogenisation done as pre-processing in the proof of Proposition C.2. The growth of constants can be analysed like blowup of \( \varepsilon \)-degree discussed in the order analysis.