Lifting Polynomial Complexity Measures Using Error-Correcting Codes

Ivan Hu
University of Wisconsin-Madison, WI, USA

Dieter van Melkebeek
University of Wisconsin-Madison, WI, USA

Abstract

We describe a method that lifts an arbitrary polynomial f with sparsity s to a polynomial that requires a read-once oblivious algebraic program of width s for every variable order. To do so, we introduce a technique for constructing a gadget based on erasure codes over finite fields, where each variable in f is substituted with a monomial that encodes a codeword into the exponents of the monomial. Our construction represents the first use of error-correcting codes in the context of lifting. As an application, we consider factor complexity, which studies how much the complexity of a polynomial can increase under factorization. Our result allows us to lift any gap in sparsity to the same gap in width in a generic manner.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases lifting, lower bound, error-correcting code, read-once oblivious algebraic branching program, polynomial factorization

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2026.64

Category RANDOM

Funding This material is based upon work supported by the National Science Foundation under Grant Nos. 2137424 and 2312540.

Acknowledgements The authors thank Nutan Limaye for useful discussions, in particular for background on algebraic proof systems, and the anonymous reviewers for helpful suggestions.

1 Introduction

The concept of lifting involves taking a function that is hard for some computational model and transforming it in some fixed manner into a different function, such that this new function is guaranteed to be hard for a stronger computational model. Lifting has proven to be a useful concept in establishing lower bounds for several computational models [26, 15, 8], proof systems [9, 30, 7, 14, 20], as well as in communication complexity [16, 17, 24, 10]. Lifting is generally done with *gadgets*, which substitute each variable in the function with an expression in some new variables. In our setting, we consider lifting in terms of the complexity of polynomials in algebraic models of computation.

We consider two measures of complexity for a polynomial f. The first notion is *sparsity*, which is simply the number of monomials needed to write f as a linear combination of those monomials. The second notion is the minimum width needed to compute f with a *read-once oblivious algebraic branching program* (roABP), which is an algebraic analogue of read-once branching programs in the Boolean setting. The model is heavily studied in contexts such as polynomial identity testing [27, 13, 12, 1, 19, 18, 6] and algebraic lower bounds [25, 2, 23]. A structural feature of roABPs is that they read variables in a particular order, and the complexity of an roABP can vary depending on this variable order. Still, roABP width is always less than or equal to the sparsity no matter the variable order, making it a stronger complexity measure.

Results. An efficient gadget for lifting a high sparsity polynomial to a polynomial requiring high roABP width has been constructed before [14, 20, 3], but only for certain variable orders for the roABP. An "intriguing question" raised in [3] is whether there is such a construction that produces a polynomial requiring high roABP width for *all* variable orders. In this work, we answer the question in the affirmative.

▶ **Theorem 1** (main result). *There exists an explicit gadget $\mathcal{G} : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[y_1, \ldots, y_{2n}]$ such that for any polynomial $f(x_1, \ldots, x_n)$ of degree d and sparsity s, $\mathcal{G}(f)$ has degree $O(nd(n+d))$, sparsity s, and requires* roABP *width s in every variable order.*

Lifting results in the algebraic setting have consequences for closure properties, in particular for factor complexity: Given a polynomial f that is efficiently computable in some model, are all of the factors of f efficiently computable in the same model? A long line of work has established positive answers for reasonably strong models of algebraic computation, starting with general algebraic circuits [22], and later branching programs [29], formulas and constant-depth circuits [5], among others. However, in the weaker model of sparsity, there exists an explicit family of polynomials with a quasi-polynomial gap between the complexity of f and a factor of f [31] over any fixed field, and one with an exponential gap with specific conditions on the underlying fields [4]. The quasi-polynomial gap for sparsity was transformed into a quasi-polynomial gap for roABPs in [3] by exploiting the specific structure of the family. With our general lifting technique, any gap in factor complexity for sparsity generically translates into the same gap for roABPs.

Technique. The algebraic setting presents new opportunities for gadgets beyond the Boolean setting. For one thing, any Boolean function can be expressed as a multilinear polynomial over $\mathbb{F}_2$, but polynomials in general can have monomials with degrees that are arbitrarily large nonnegative integers. This allows us to encode information into the exponents of a monomial.

Our main idea is to use "codeword monomials" to replace each variable of the original polynomial f. More specifically, our gadget $\mathcal{G}$ substitutes each variable x_i with a monomial in new variables $y_1, \ldots, y_\ell$. A monomial $x_1^{d_1} \cdots x_n^{d_n}$ in f can be thought of as a vector $[d_1, \ldots, d_n] \in \mathbb{Z}^n$. The effect of the gadget can be interpreted as a linear map from $\mathbb{Z}^n \to \mathbb{Z}^\ell$ acting on each monomial of f. Our gadget essentially is a linear error-correcting code $E_p : \mathbb{F}_p^n \to \mathbb{F}_p^\ell$ for some sufficiently large prime p, where the elements of $\mathbb{F}_p$ are interpreted as elements of $\mathbb{Z}$ to serve as exponents.

How do we ensure that the sparsity of f becomes a lower bound for the roABP width of $\mathcal{G}(f)$ in every variable order? We use the well-known characterization of roABP width for a fixed variable order as the maximum rank of its coefficient matrices. Each coefficient matrix consists of the coefficients of the monomials of f up to the degree of f, arranged in a particular way. The injectivity of E guarantees that each monomial of f is mapped to a distinct monomial of $\mathcal{G}(f)$. By choosing E judiciously, we additionally ensure that for at least one of the coefficient matrices of $\mathcal{G}(f)$ in every order, the entries corresponding to those distinct monomials have distinct row indices and distinct column indices. Thus, if f has exactly s monomials, the s nonzero entries of this coefficient matrix are spread over distinct rows and distinct columns, which ensures that the matrix rank is s. As E_p is independent of the variable order, this yields a polynomial $\mathcal{G}(f)$ that requires roABP width at least s in every possible variable order.

We show that an erasure code E_p with appropriate parameters achieves our goal. The structure of the various coefficient matrices of $\mathcal{G}(f)$ depends on a partition of the variables $\{y_1, \ldots, y_\ell\}$ into two disjoint sets U and V. The row of the entry controlled by a given

monomial $\overline{m}$ is determined by the U-variables in $\overline{m}$, while the column is determined by the V-variables. With an erasure code gadget that can handle $|V|$ erasures, any $\overline{m}$ produced by $\mathcal{G}$ can be uniquely recovered from just knowing the U-variables. The effect is that no two nonzero entries share the same row. By swapping the roles of U and V, a similar property follows for columns instead of rows. An error-correcting code $E_p : \mathbb{F}_p^n \to \mathbb{F}_p^\ell$ that can handle erasures of up to half its output length ℓ has both properties for $|U| = |V| = \ell/2$. Reed-Solomon codes can handle the required number of erasures with $\ell = 2n$.

Organization. We start with the required background in Section 2, develop our results from first principles in Section 3, and end in Section 4 with directions for further research.

2 Preliminaries

We review elements of coding theory and the notions of polynomial sparsity and roABP's. As a warm-up, we also include a proof of the known gadget that works for restricted orders only. Throughout the paper, we use the notation $[n] = \{1, \ldots, n\}$ for positive integers n.

2.1 Codes

For a prime p, let $\mathbb{F}_p$ be the finite field of order p. For positive integers k, ℓ, a *linear error-correcting code* over $\mathbb{F}_p$ of dimension k and length ℓ is an injective linear map $E_p : \mathbb{F}_p^k \to \mathbb{F}_p^\ell$. We write $C \doteq E_p(\mathbb{F}_p^k)$ to denote the set of *codewords* of E_p. We say that E_p can *handle* e *erasures* if deleting up to e positions in a codeword of E_p still leaves enough information to retrieve the codeword. In symbols, for any two codewords $\mathbf{c}, \mathbf{c}' \in C$, and any subset of indices $I \subseteq [\ell]$ with $|I| \geq \ell - e$, we have that if $\mathbf{c}|_I = \mathbf{c}'|_I$, then $\mathbf{c} = \mathbf{c}'$.

We recall the properties of the classical Reed-Solomon code that we need. For completeness, we include the construction and proof.

▶ **Lemma 2** (Reed-Solomon, [28]). *For any positive integers k, e and prime $p \geq k + e$, there exists a linear code E_p over $\mathbb{F}_p$ with dimension k and length $k + e$ that can handle e erasures.*

Proof. Let $\alpha_1, \ldots, \alpha_{k+e}$ be fixed distinct elements in $\mathbb{F}_p$. For any vector $\mathbf{u} = \begin{bmatrix} u_1 & \ldots & u_k \end{bmatrix}$ in $\mathbb{F}_p^k$, define $f_{\mathbf{u}}(x) \in \mathbb{F}_p[x]$ to be the degree $k-1$ univariate polynomial

$$f_{\mathbf{u}}(x) = u_1 + u_2 x + \cdots + u_k x^{k-1}.$$

We define E as follows:

$$E_p(\mathbf{u}) = \begin{bmatrix} f_{\mathbf{u}}(\alpha_1) & \ldots & f_{\mathbf{u}}(\alpha_{k+e}) \end{bmatrix}.$$

We now show that E_p can handle e erasures. Let $I = \{i_1, \ldots, i_k\} \subseteq [k+e]$ be a set of $(k+e) - e = k$ indices. Suppose there are two codewords $\mathbf{c} = E_p(\mathbf{u})$, $\mathbf{c}' = E_p(\mathbf{u}')$ such that $\mathbf{c}|_I = \mathbf{c}'|_I$. Then for $j \in [k]$, we have that $f_{\mathbf{u}}(\alpha_{i_j}) = f_{\mathbf{u}'}(\alpha_{i_j})$. In other words, $f_{\mathbf{u}}$ and $f_{\mathbf{u}'}$ are degree $k-1$ polynomials that are equal at k distinct values. This implies that $f_{\mathbf{u}} = f_{\mathbf{u}'}$, meaning $\mathbf{c} = \mathbf{c}'$. ◀

Any element of $\mathbb{F}_p$ can be interpreted as an integer in $[0, p-1]$. For a linear code $E_p : \mathbb{F}_p^k \to \mathbb{F}_p^\ell$, we write the function $\widetilde{E}_p : \mathbb{F}_p^k \to \mathbb{Z}^\ell$ by interpreting the outputs of E_p as integers in this manner. Note that $\widetilde{E}_p$ is not a linear map, but if $E_p(\mathbf{u}) = \sum_{i=1}^s c_i E_p(\mathbf{v}_i)$, then $\widetilde{E}_p(\mathbf{u}) \equiv \sum_{i=1}^s c_i \widetilde{E}_p(\mathbf{v}_i) \bmod p$. Also, we abuse notation to allow vectors in $\mathbb{Z}^k$ as arguments to E_p and $\widetilde{E}_p$ by taking the entries modulo p.

2.2 Polynomials

Let $\mathbb{F}$ be a field (which may be unrelated to $\mathbb{F}_p$), and let $X = \{x_1, \ldots, x_n\}$ be a set of variables. A *monomial* $\overline{m}$ in X is a product of not necessarily distinct variables in X. For a subset $U \subseteq X$, define $\overline{m}_U$ to be the monomial produced by taking $\overline{m}$ and only including the variables in U (equivalently, setting $x_i = 1$ for $x_i \notin U$).

We can write a monomial uniquely as $\overline{m} = x_1^{d_1} \cdots x_n^{d_n}$, where $d_1, \ldots, d_n \geq 0$ are integers. We call the vector $[d_1, \ldots, d_n]$ the *degree vector* of $\overline{m}$. We refer to $\max_i(d_i)$ as the *individual degree* of $\overline{m}$, and to $\deg(\overline{m}) = \sum_i d_i$ as the *(total) degree* of $\overline{m}$. The individual and total degrees of a polynomial are defined as the maximum of the respective degrees of its monomials.

▶ **Definition 3** (sparsity). *For a polynomial $f \in \mathbb{F}[X]$, the sparsity of f is the number of terms used when f is written as a linear combination of monomials in X.*

2.3 Read-once oblivious algebraic branching programs

A *read-once oblivious algebraic branching program* (roABP) consists of a directed graph with $n+1$ layers labeled $0, 1, \ldots, n$. Layer 0 contains a source vertex s, layer n contains a sink vertex t, and each edge goes from a layer $i-1$ to layer i, for $i = 1, \ldots, n$. The other parameters are a set of n variables $x_1, \ldots, x_n$ and a variable order, which is a permutation $\pi : [n] \to [n]$. Each edge from layer $i-1$ to i is labeled with a univariate polynomial in $x_{\pi(i)}$. For every path γ from s to t, let the weight $\mathrm{wt}(\gamma)$ be the product of all the labels of the edges in γ. The polynomial f computed by the roABP is given by the sum of all the path weights from s to t:

$$f \doteq \sum_{\gamma : s \rightsquigarrow t} \mathrm{wt}(\gamma).$$

An roABP computes f only if both sides in the above equality are identical as polynomials in $\mathbb{F}[X]$, as opposed to just having the equality holding upon evaluation on all points in $\mathbb{F}^n$. If $|\mathbb{F}| > \deg(f)$, both notions are equivalent.

The *depth* of an roABP is the number of edge layers in the graph, i.e., n. The *width* of an roABP is equal to the maximum number of vertices in any layer. The *size* of an roABP is the total number of vertices in the graph. The primary measure of complexity of an roABP is width, since the depth is simply the number of variables, and the size is bounded by the product of width and depth.

▶ **Definition 4** (roABP width). *For a polynomial $f \in \mathbb{F}[X]$ and permutation $\pi : [n] \to [n]$, $\mathsf{roABP\text{-}width}_\pi(f)$ denotes the minimum width of any* roABP *of variable order π that computes f. $\mathsf{roABP\text{-}width}(f)$ denotes the minimum value of $\mathsf{roABP\text{-}width}_\pi(f)$ over all variable orders π.*

In our work, we do not directly use this definition to determine roABP width. Instead, we use a well-known characterization by Nisan [25] for non-commutative ABPs, extended to roABPs in [11, Lemma 4.5.8]. It expresses $\mathsf{roABP\text{-}width}_\pi(f)$ in terms of the ranks of its coefficient matrices.

▶ **Definition 5** (coefficient matrix). *Consider a partition $X \doteq U \sqcup V$. For a polynomial $f \in \mathbb{F}[X]$, we define a matrix $\mathrm{CMat}_{U,V}(f)$ as follows. The rows are indexed by monomials $\overline{m}_U$ over U, and the columns are indexed by monomials $\overline{m}_V$ over V. The entry of $\mathrm{CMat}_{U,V}(f)$ indexed by $(\overline{m}_U, \overline{m}_V)$ is the coefficient of $\overline{m}_U \overline{m}_V$ in f.*

Since we are only considering monomials of degree at most $\deg(f)$, we can assume that the dimensions of $\mathrm{CMat}_{U,V}(f)$ are finite.

▶ **Lemma 6** (roABP width characterization). *Let $f \in \mathbb{F}[X]$ be a polynomial, and let $\pi : [n] \to [n]$ be a variable order. For each $i \in \{0, \ldots, n\}$, let $U_i = \{x_{\pi(1)}, \ldots, x_{\pi(i)}\}$ and $V_i = \{x_{\pi(i+1)}, \ldots, x_{\pi(n)}\}$ be a partition of X. Then*

$$\mathsf{roABP\text{-}width}_\pi(f) = \max_{i \in \{0,\ldots,n\}} \operatorname{rank}(\mathrm{CMat}_{U_i,V_i}(f)).$$

roABP width is a stronger measure of complexity than sparsity. Specifically, any polynomial f with sparsity s satisfies $\mathsf{roABP\text{-}width}(f) \leq s$. To see this, one can construct an roABP with s disjoint paths or use Lemma 6 to see that any coefficient matrix has at most s nonzero entries, and thus has rank at most s.

The roABP width of a polynomial can vary wildly based on the variable order π. For example, consider the polynomial $f = \prod_{i=1}^{n}(x_i + y_i)$. If the variables are read in the order $x_1, y_1, \ldots, x_n, y_n$, then $\mathsf{roABP\text{-}width}_\pi(f) = 2$, while if the variables are read in the order $x_1, \ldots, x_n, y_1, \ldots, y_n$, then $\mathsf{roABP\text{-}width}_\pi(f) = 2^n$.

2.4 Lifting from sparsity to fixed-order roABP width

As a warm-up, we present a well-known gadget for lifting from sparsity to roABP width with respect to particular variable orders. Our gadget does not rely on it, but the construction was used in [14, 20, 3].

▶ **Proposition 7.** *There exists an explicit gadget $\mathcal{G} : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[y_1, \ldots, y_n, z_1, \ldots, z_n]$ such that for any polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ of total degree d and sparsity s, $\mathcal{G}_0(f)$ has the same individual degree at f, total degree $2d$, and $\mathsf{roABP\text{-}width}_\pi$ equal to s for every variable order π where the variables $y_1, \ldots, y_n$ are read before the variables $z_1, \ldots, z_n$.*

Proof. We define $\mathcal{G}(x_i) = y_i z_i$ for each $i \in [n]$, and write $g = \mathcal{G}(f)$. The individual degree of y_i and z_i in g are the same as the individual degree of x_i in f. The total degree of every monomial doubles, so $\deg(g) = 2d$.

For the roABP width, partition the variables $\{y_1, \ldots, y_n, z_1, \ldots, z_n\}$ into $Y = \{y_1, \ldots, y_n\}$, $Z = \{z_1, \ldots, z_n\}$. Notice that for any monomial $\overline{m}$ appearing in g, $\overline{m}_Y$ uniquely determines $\overline{m}_Z$ and vice versa, since $\overline{m}_Y$ and $\overline{m}_Z$ have the same degree vectors. Therefore, $\mathrm{CMat}_{Y,Z}(g)$ is a matrix with at most one nonzero entry in each row and column, and the number of nonzero entries is s. So $\operatorname{rank}(\mathrm{CMat}_{Y,Z}(g)) = s$ and Lemma 6 implies $\mathsf{roABP\text{-}width}_\pi(g) \geq s$. To see that $\mathsf{roABP\text{-}width}_\pi(g) \leq s$, notice that $\mathcal{G}$ maps monomials to monomials and thus g has sparsity at most s. ◀

3 Results

In this section we present our main result and its consequences for factor complexity.

3.1 Main result

Earlier gadgets for lifting from sparsity to roABP width either only work for restricted orders or require structural properties of the polynomials being lifted. We remedy these shortcomings and develop a gadget that works for every order and every polynomial.

Let $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_{2n}\}$ be sets of variables. We define the gadget $\mathcal{G} : \mathbb{F}[X] \to \mathbb{F}[Y]$ for Theorem 1 as follows. For a prime $p > 2n$, let $E_p : \mathbb{F}_p^n \to \mathbb{F}_p^{2n}$ be a linear error-correcting code that can handle n erasures. Such a code exists by virtue of Lemma 2. Let $\widetilde{E}_p : \mathbb{F}_p^n \to \mathbb{Z}^{2n}$ be the corresponding integer-valued function. We write $\mathbf{e}_1, \ldots, \mathbf{e}_n \in \mathbb{F}_p^n$ as the standard basis vectors. We define $\mathcal{G}_p$ to perform the following substitution on a variable x_i.

$$\mathcal{G}_p(x_i) = \prod_{j=1}^{2n} y_j^{\widetilde{E}_p(\mathbf{e}_i)_j}. \tag{1}$$

Notice that if $\overline{m}$ is a monomial in X, then $\mathcal{G}_p(\overline{m})$ is a monomial in Y. The erasure capability of E allows us to prove the following structural lemma.

▶ **Lemma 8.** *Let n, d be positive integers, and $p > \max(2n, d)$ a prime. Let $Y \doteq U \sqcup V$ be any partition such that $|U| = |V| = n$. Let $\overline{m}_1, \overline{m}_2$ be monomials of individual degree at most d in X. If $\overline{m}_1 \neq \overline{m}_2$, then $\mathcal{G}_p(\overline{m}_1)_U \neq \mathcal{G}_p(\overline{m}_2)_U$ and $\mathcal{G}_p(\overline{m}_1)_V \neq \mathcal{G}_p(\overline{m}_2)_V$.*

Proof. The condition that $p > 2n$ ensures that the gadget $\mathcal{G}_p$ is well-defined.

Let $\overline{m}$ be a monomial in X with degree vector $\mathbf{d} = [d_1, \ldots, d_n]$. Since $d_1 E_p(\mathbf{e}_1) + \cdots + d_n E_p(\mathbf{e}_n) = E_p(\mathbf{d})$, we have that $d_1 \widetilde{E}_p(\mathbf{e}_1) + \cdots + d_n \widetilde{E}_p(\mathbf{e}_n) \equiv \widetilde{E}_p(\mathbf{d}) \bmod p$. In other words, there exist integers $q_1, \ldots, q_{2n}$ such that

$$\mathcal{G}_p(\overline{m}) = \prod_{j=1}^{2n} y_j^{d_1 \widetilde{E}_p(\mathbf{e}_1)_j + \cdots + d_n \widetilde{E}_p(\mathbf{e}_n)_j} = \prod_{j=1}^{2n} y_j^{\widetilde{E}_p(\mathbf{d})_j + pq_j}.$$

Armed with this fact, let $\overline{m}_1, \overline{m}_2$ be monomials of degree at most d in X. Suppose $\overline{m}_1, \overline{m}_2$ have degree vectors $\mathbf{d}_1$, $\mathbf{d}_2$, respectively. Furthermore, suppose that for $U = \{y_{i_1}, \ldots, y_{i_n}\} \subseteq Y$, we have that $\mathcal{G}_p(\overline{m}_1)_U = \mathcal{G}_p(\overline{m}_2)_U$. In other words, the powers of the U-variables in $\mathcal{G}_p(\overline{m}_1)$ and $\mathcal{G}_p(\overline{m}_2)$ agree. This means that, for $j \in [n]$, $\widetilde{E}_p(\mathbf{d}_1)_{i_j} \equiv \widetilde{E}_p(\mathbf{d}_2)_{i_j} \bmod p$, or $E_p(\mathbf{d}_1)_{i_j} = E_p(\mathbf{d}_2)_{i_j}$. So, $E_p(\mathbf{d}_1)$ and $E_p(\mathbf{d}_2)$ agree on at least n indices. Because E_p has output length $2n$ and can handle n erasures, we have that $\mathbf{d}_1 \equiv \mathbf{d}_2 \bmod p$. Since $p > d$ and each entry of $\mathbf{d}_1, \mathbf{d}_2$ is at most d, we have that $\mathbf{d}_1 = \mathbf{d}_2$. Therefore, if $\mathcal{G}_p(\overline{m}_1)_U = \mathcal{G}_p(\overline{m}_2)_U$, then $\overline{m}_1 = \overline{m}_2$.

We can conclude that if $\overline{m}_1 \neq \overline{m}_2$, then $\mathcal{G}_p(\overline{m}_1)_U \neq \mathcal{G}_p(\overline{m}_2)_U$. By the same reasoning, we can see that $\mathcal{G}_p(\overline{m}_1)_V \neq \mathcal{G}_p(\overline{m}_2)_V$. ◀

Now we show that our gadget $\mathcal{G}_p$ satisfies the conditions of Theorem 1. In fact, we prove a refined statement with Theorem 9 that considers the individual degree of each variable.

▶ **Theorem 9** (refined main result). *There exists an explicit gadget $\mathcal{G} : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[y_1, \ldots, y_{2n}]$ such that for any polynomial $f(x_1, \ldots, x_n)$ of degree d and sparsity s, $\mathcal{G}(f)$ has individual degree $O(d(n+d))$ in each variable, sparsity s, and* roABP-width$(\mathcal{G}(f)) = s$.

Proof. Let $m \doteq \max(2n, d) + 1$. By Bertrand's postulate, there always exists a prime $p \in [m, 2m]$. We show that $\mathcal{G} \doteq \mathcal{G}_p$ satisfies the conditions of the theorem.

Write $g = \mathcal{G}(f)$. First, we prove the degree bound. Since $\widetilde{E}_p(\mathbf{e}_i)_j < p \leq 2m$ for any $i \in [n]$, $j \in [2n]$, $\mathcal{G}(x_i)$ is a monomial in Y of individual degree less than $2m$. As each monomial in f contains at most d occurrences of variables (counting multiplicities), after replacing each x_i by $\mathcal{G}(x_i)$, the resulting monomials have individual degree less than $md = O(d(n+d))$.

Now we show the lower bound on roABP width. Let $\pi : [2n] \to [2n]$ be any variable order on Y. Let $U = \{y_{\pi(1)}, \ldots, y_{\pi(n)}\}$ and $V = \{y_{\pi(n+1)}, \ldots, y_{\pi(2n)}\}$. We reason about the structure of the coefficient matrix $\mathrm{CMat}_{U,V}(g)$. Since f has sparsity s, we can write it as a linear combination of s monomials: $f = \sum_{i=1}^{s} c_i \overline{m}_i$ for some $c_i \neq 0$. We have

$$\mathcal{G}(f) = \sum_{i=1}^{s} c_i \mathcal{G}(\overline{m}_i) = \sum_{i=1}^{s} c_i \mathcal{G}(\overline{m}_i)_U \mathcal{G}(\overline{m}_i)_V.$$

By Lemma 8, all monomials in the sets $\{\mathcal{G}(\overline{m}_1)_U,\ldots,\mathcal{G}(\overline{m}_s)_U\}$ and $\{\mathcal{G}(\overline{m}_1)_V,\ldots,\mathcal{G}(\overline{m}_s)_V\}$ are distinct. Thus, for $i \in [s]$, the entry $(\mathcal{G}(\overline{m}_i)_U, \mathcal{G}(\overline{m}_i)_V)$ of $\mathrm{CMat}_{U,V}(g)$ is nonzero, all other entries of $\mathrm{CMat}_{U,V}(g)$ are zero, and no two of the s nonzero entries share a row or column. This implies that $\mathrm{rank}(\mathrm{CMat}_{U,V}(g)) \geq s$, and by Lemma 6, this implies that $\mathsf{roABP\text{-}width}_\pi(g) \geq s$. Since π was arbitrary, we have that $\mathsf{roABP\text{-}width}(g) \geq s$.

Finally, since $\mathcal{G}$ maps monomials to monomials, g still has sparsity s, so there exists an roABP of width s computing g, no matter the variable order. Therefore, $\mathsf{roABP\text{-}width}(g) = s$. ◀

3.2 Application to factor complexity

We now establish the consequences for factor complexity that we mentioned in the introduction. They immediately follow from our main result and two well-known gaps for sparsity.

Over any field, the following gap in sparsity was established in [31].

▶ **Fact 10.** For all fields $\mathbb{F}$ and positive integers n and d, there exists a degree-nd polynomial $f(x_1,\ldots,x_n)$ of sparsity 2^n that has a factor $g(x_1,\ldots,x_n)$ of sparsity d^n. Specifically,

$$f = \prod_{i=1}^{n}(x_i^d - 1) \text{ and } g = \prod_{i=1}^{n}(1 + x_i + x_i^2 + \cdots + x_i^{d-1}).$$

Over a field of nonzero characteristic d, the identity $(x_1 + \cdots + x_n)^d = x_1^d + \cdots + x_n^d$ yields the following gap (see, e.g., [4, Example 4.7]).

▶ **Fact 11.** Suppose $d = \mathrm{char}(\mathbb{F}) \neq 0$. For all positive integers n, there exists a degree-d polynomial $f(x_1,\ldots,x_n)$ of sparsity n that has a factor $g(x_1,\ldots,x_n)$ of sparsity $\binom{n+d-2}{d-1}$. Specifically,

$$f = x_1^d + \cdots + x_n^d \text{ and } g = (x_1 + \cdots + x_n)^{d-1}.$$

Notice that by the multinomial theorem and the fact that d is prime, every monomial of degree up to $d-1$ occurs in g with a nonzero coefficient modulo d.

By lifting Fact 10 and Fact 11 using our main result, we obtain the following results for roABP width.

▶ **Corollary 12.** *Let $\mathbb{F}$ be a field and n a positive integer.*

(a) *For every positive integer d, there exists a degree-$O(d^2n^3)$ polynomial $F(y_1,\ldots,y_{2n})$ such that $\mathsf{roABP\text{-}width}(F)=2^n$, and F has a factor $G(y_1,\ldots,y_{2n})$ such that $\mathsf{roABP\text{-}width}(G) = d^n$.*

(b) *If $\mathrm{char}(\mathbb{F}) = d \neq 0$, then there exists a degree-$O(nd(n+d))$ polynomial $F(y_1,\ldots,y_{2n})$ such that $\mathsf{roABP\text{-}width}(F) = n$, and F has a factor $G(y_1,\ldots,y_{2n})$ such that $\mathsf{roABP\text{-}width}(G) = \binom{n+d-2}{d-1}$.*

Proof. Since $\mathcal{G}$ is a substitution, if g is a factor of f, then $G \doteq \mathcal{G}(g)$ is a factor of $F \doteq \mathcal{G}(f)$. Apply Theorem 1 to Fact 10 and Fact 11. Respectively, we obtain $\mathsf{roABP\text{-}width}(F) = 2^n$ and $\mathsf{roABP\text{-}width}(G) = d^n$, and $w \doteq \mathsf{roABP\text{-}width}(F) = n$ and $\mathsf{roABP\text{-}width}(G) = \binom{n+d-2}{d-1}$. ◀

Note that for fixed d, the width gap in part (a) of Corollary 12 goes from $w = 2^n$ to $d^n = w^{\Omega(\log d)}$, and in part (b) from $w = n$ to $\binom{n+d-2}{d-1} = w^{\Omega(d)}$. By instead setting $d = n$, we obtain an explicit family of polynomials with a quasi-polynomial gap $w \mapsto w^{\Omega(\log w)}$ in case (a), and an exponential gap $w \mapsto \binom{2w-2}{w-1} = \Theta(2^{2w}/\sqrt{w})$ in case (b).

In comparison to the approach in [3], our technique has the advantage of exactly preserving the complexity measure, while [3] weakens the complexity of the factor slightly. Furthermore, we can apply our methods to any family of polynomials, while [3] requires specific conditions on the family. On the other hand, [3] has the advantage that the degree of the polynomial increases by a constant factor, while our gadget increases the degree by a factor of $O(n(n+d))$.

4 Further research

Error-correcting codes have been helpful in many contexts. We exhibit their first use in the context of lifting and manage to solve an open problem in lifting raised by [3]. Can our technique be applied to other lifting results, as well? For example, does it yield generic lifting lower bounds for roABPs in the context of algebraic proof systems such as fragments of the Ideal Proof System (IPS)? Previous works in this area [14, 20, 21] established lower bounds for specific instances of roABP-IPS. An interesting open question is to construct a gadget that takes a generic hard instance for a weaker fragment of IPS and lifts it to an instance that is hard for roABP-IPS.

On a more technical level, in comparison to other examples of lifting like [3], the gadget in our work increases the degree of the polynomial by a relatively large factor, polynomial in the original degree d and number of variables n. Could one construct a more efficient gadget that lifts sparsity to roABP width? For example, can one construct a gadget that only uses multilinear codeword monomials?

In the context of factor gap complexity, for degree-d polynomials of sparsity s, the best known gaps are essentially the ones given in Section 3.2, namely that a factor can have sparsity $s^{\Omega(\log d)}$ over any field and $s^{\Omega(d)}$ over fields with nonzero characteristic d. [4] showed an upper bound of $s^{O(d^2 \log n)}$ for the sparsity of any factor over any field, where n is the number of variables. Can one construct examples that achieve larger gaps in sparsity and therefore, by our main result, in roABPs width? Alternatively, can one find larger gaps in roABPs width independent of any results on sparsity?

References

1 Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-sets for ROABP and sum of set-multilinear circuits. *SIAM J. Comput.*, 44(3):669–697, 2015. doi:10.1137/140975103.

2 Matthew Anderson, Michael A. Forbes, Ramprasad Saptharishi, Amir Shpilka, and Ben Lee Volk. Identity testing and lower bounds for read-k oblivious algebraic branching programs. *ACM Trans. Comput. Theory*, 10(1):3:1–3:30, 2018. doi:10.1145/3170709.

3 Robert Andrews, Jules Armand, Prateek Dwivedi, Magnus Rahbek Dalgaard Hansen, Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. On closure properties of read-once oblivious algebraic branching programs. In *17th Innovations in Theoretical Computer Science Conference, ITCS 2026, Bocconi University, Milan, Italy, January 27-30, 2026*, LIPIcs, pages 9:1–9:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026. doi:10.4230/LIPIcs.ITCS.2026.9.

4 Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. Deterministic factorization of sparse polynomials with bounded individual degree. *J. ACM*, 67(2):8:1–8:28, 2020. doi:10.1145/3365667.

5 Somnath Bhattacharjee, Mrinal Kumar, Shanthanu S. Rai, Varun Ramanathan, Ramprasad Saptharishi, and Shubhangi Saraf. Closure under factorization from a result of Furstenberg. *CoRR*, abs/2506.23214, 2025. doi:10.48550/arXiv.2506.23214.

6 Pranav Bisht and Nitin Saxena. Blackbox identity testing for sum of special ROABPs and its border class. *Comput. Complex.*, 30(1):8, 2021. doi:10.1007/S00037-021-00209-Y.

7 Yogesh Dahiya, Meena Mahajan, and Sasank Mouli. New lower bounds for polynomial calculus over non-Boolean bases. In *27th International Conference on Theory and Applications of Satisfiability Testing, SAT 2024, Pune, India, August 21-24, 2024*, LIPIcs, pages 10:1–10:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.SAT.2024.10.

8 Susanna F. de Rezende, Mika Göös, and Robert Robere. Guest column: Proofs, circuits, and communication. *SIGACT News*, 53(1):59–82, 2022. doi:10.1145/3532737.3532746.

9 Susanna F. de Rezende, Or Meir, Jakob Nordström, Toniann Pitassi, Robert Robere, and Marc Vinyals. Lifting with simple gadgets and applications to circuit and proof complexity. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 24–30. IEEE, 2020. doi:10.1109/FOCS46700.2020.00011.

10 Susanna F. de Rezende and Marc Vinyals. Lifting with colourful sunflowers. In *40th Computational Complexity Conference, CCC 2025, Toronto, Canada, August 5-8, 2025*, LIPIcs, pages 36:1–36:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.CCC.2025.36.

11 Michael A. Forbes. *Polynomial identity testing of read-once oblivious algebraic branching programs*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2014. URL: https://hdl.handle.net/1721.1/89843.

12 Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 867–875. ACM, 2014. doi:10.1145/2591796.2591816.

13 Michael A. Forbes and Amir Shpilka. Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, Berkeley, CA, USA, October, 26-29, 2013*, pages 243–252. IEEE Computer Society, 2013. doi:10.1109/FOCS.2013.34.

14 Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. Proof Complexity Lower Bounds from Algebraic Circuit Complexity. *Theory of Computing*, 17:1–88, 2021. doi:10.4086/TOC.2021.V017A010.

15 Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. *Theory Comput.*, 16:1–30, 2020. doi:10.4086/TOC.2020.V016A013.

16 Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. *SIAM J. Comput.*, 47(6):2435–2450, 2018. doi:10.1137/16M1059369.

17 Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. *SIAM J. Comput.*, 49(4), 2020. doi:10.1137/17M115339X.

18 Zeyu Guo and Rohit Gurjar. Improved explicit hitting-sets for roabps. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, Virtual Conference, August 17-19, 2020*, LIPIcs, pages 4:1–4:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.APPROX/RANDOM.2020.4.

19 Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf. Deterministic identity testing for sum of read-once oblivious arithmetic branching programs. *Comput. Complex.*, 26(4):835–880, 2017. doi:10.1007/S00037-016-0141-Z.

20 Tuomas Hakoniemi, Nutan Limaye, and Iddo Tzameret. Functional lower bounds in algebraic proofs: Symmetry, lifting, and barriers. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 1396–1404. ACM, 2024. doi:10.1145/3618260.3649616.

21 Tuomas Hakoniemi, Nutan Limaye, and Iddo Tzameret. Hard CNF instances for ideal proof systems. *CoRR*, abs/2605.04544, 2026. doi:10.48550/arXiv.2605.04544.

22 Erich L. Kaltofen. Factorization of polynomials given by straight-line programs. *Adv. Comput. Res.*, 5:375–412, 1989.

23 Neeraj Kayal, Vineet Nair, and Chandan Saha. Separation between read-once oblivious algebraic branching programs (ROABPs) and multilinear depth-three circuits. *ACM Trans. Comput. Theory*, 12(1):2:1–2:27, 2020. doi:10.1145/3369928.

24 Shachar Lovett, Raghu Meka, Ian Mertz, Toniann Pitassi, and Jiapeng Zhang. Lifting with sunflowers. In *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, Berkeley, CA, USA, January 31 - February 3, 2022*, LIPIcs, pages 104:1–104:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ITCS.2022.104.

25 Noam Nisan. Lower bounds for non-commutative computation. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 410–418. ACM, 1991. doi:10.1145/103418.103462.

26 Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Comb.*, 19(3):403–435, 1999. doi:10.1007/S004930050062.

27 Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Comput. Complex.*, 14(1):1–19, 2005. doi:10.1007/S00037-005-0188-8.

28 Irving S. Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960. doi:10.1137/0108018.

29 Amit Sinhababu and Thomas Thierauf. Factorization of polynomials given by arithmetic branching programs. *Computational Complexity*, 30(2):15:1–15:47, 2021. doi:10.1007/S00037-021-00215-0.

30 Dmitry Sokolov. (Semi)Algebraic proofs over $\{\pm 1\}$ variables. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 78–90. Association for Computing Machinery, 2020. doi:10.1145/3357713.3384288.

31 Joachim von zur Gathen and Erich L. Kaltofen. Factoring sparse multivariate polynomials. *J. Comput. Syst. Sci.*, 31(2):265–287, 1985. doi:10.1016/0022-0000(85)90044-3.