Electronic Colloquium on Computational Complexity, Report No. 125 (2026)

Lifting Polynomial Complexity Measures Using Error-Correcting Codes

Ivan Hu Dieter van Melkebeek

July 24, 2026

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. To the best of our knowledge, 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.

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 [RM99, GGKS20, dRGR22], proof systems [dRMN$^+$20, Sok20, DMM24, FSTW21, HLT24], as well as in communication complexity [GPW18, GPW20, LMM$^+$22, dRV25]. 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 [RS05, FS13, FSS14, AGKS15, GKST17, GG20, BS21] and algebraic lower bounds [Nis91, AFS$^+$18, KNS20]. 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 regardless of the variable order, making it a stronger complexity measure.

Results. An explicit gadget for lifting a high sparsity polynomial to a polynomial requiring high roABP width has been constructed before [FSTW21, HLT24, AAD$^+$26], but only for certain

ISSN 1433-8092

variable orders for the roABP. An "intriguing question" raised in [AAD+26] 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 $\mathcal{G}$ that maps every n-variate polynomial f of degree d and sparsity s to a $(2n)$-variate polynomial $\mathcal{G}(f)$ of degree $O(nd(n+d))$ and sparsity s such that $\mathcal{G}(f)$ 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 [Kal89], and later branching programs [ST21], formulas and constant-depth circuits [BKR+25], 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 an irreducible factor of f over any fixed field [vK85] and one with an exponential gap under certain conditions on the underlying fields [BSV20]. The quasi-polynomial gap for sparsity was transformed into a quasi-polynomial gap for roABP width in [AAD+26] by exploiting the specific structure of the family. With our general gadget, any gap in factor complexity for sparsity generically translates into the same gap for roABP width.

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 is essentially 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_p guarantees that each monomial of f is mapped to a distinct monomial of $\mathcal{G}(f)$. By choosing E_p 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 notions and background in Section 2, develop our results from first principles in Section 3, present earlier gadgets and compare them with ours in Section 4, and end with directions for further research in Section 5.

2 Preliminaries

We review the notions of polynomial gadget, sparsity, roABPs, and elements of coding theory. Throughout the paper, we use the notation $[n] = \{1, \dots, n\}$ for positive integers n.

2.1 Polynomials

Let $\mathbb{F}$ be a field and let $X \subseteq \{x_1, x_2, \dots\}$ be a set of variables. A *monomial* $\overline{m}$ in X is a finite 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 in $X_n \doteq \{x_1, \dots, x_n\}$ uniquely as $\overline{m} = x_1^{d_1} \cdots x_n^{d_n}$, where $d_1, \dots, d_n \geq 0$ are integers. We call the vector $[d_1, \dots, d_n]$ 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 2 (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.

The notion of a *gadget* can be formalized in terms of homomorphisms from (subsets of) $\mathbb{F}[X]$ to $\mathbb{F}[Y]$, where $Y = \{y_1, y_2, \dots\}$ represents another set of variables. The homomorphisms preserve scalar multiplication, polynomial addition, and polynomial multiplication; we refer to them as algebra homomorphisms, even though their domains may not be closed under polynomial multiplication. In particular, we consider domains where both the number of variables and the individual degree are restricted. We include upper bounds on those quantities as parameters into a family of homomorphisms, enabling our constructions to make use of those bounds.

Definition 3 (gadget). A gadget $\mathcal{G}$ is a collection of mappings $\mathcal{G}_{n,d}$, one for every positive integer n and nonnegative integer d, such that $\mathcal{G}_{n,d}$ is an algebra homomorphism from the set of polynomials in $\mathbb{F}[X_n]$ with individual degree at most d to $\mathbb{F}[Y]$.

In principle, a gadget can depend on the underlying field $\mathbb{F}$. However, typical gadgets (including ours) do not depend on $\mathbb{F}$; we call such gadgets *oblivious*.

The mapping $\mathcal{G}_{n,d}$ is uniquely determined by $\mathcal{G}_{n,d}(x_i)$ for each $i \in [n]$. In general, a gadget is called explicit if a representation of $\mathcal{G}_{n,d}(x_i)$ can be computed deterministically in time polynomial in n and d. The general notion presupposes an underlying model of computation for the representation. However, for the gadgets we consider, $\mathcal{G}_{n,d}(x_i)$ always is a monomial, which can be represented by

its degree vector. We call such a gadget *explicit* if the degree vector of $\mathcal{G}_{n,d}(x_i)$ can be computed deterministically in time polynomial in n and d, given n, d, and $i \in [n]$.

2.2 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 $X_n = \{x_1, \ldots, x_n\}$ 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 $f \in \mathbb{F}[X_n]$ 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 [Nis91] for non-commutative ABPs, extended to roABPs in [For14, 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 $X_n \doteq U \sqcup V$. For a polynomial $f \in \mathbb{F}[X_n]$, 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. Also, for specificity, we can assume the default that the monomials are considered in lexicographic order. The precise order does not matter to us as we only care about the rank of the coefficient matrices, and the rank is unaffected by reordering rows and/or columns.

Lemma 6 (roABP width characterization). *Let $f \in \mathbb{F}[X_n]$ 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*

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

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 $\mathsf{roABP\text{-}width}_\pi(f) = 2^n$.

2.3 Error-correcting 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 codes [RS60] that we need. For completeness, we include the construction and proof.

Lemma 7 (Reed-Solomon codes). *For all 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 and can be computed in time polynomial in $k+e$ and $\log(p)$.*

Proof. Let $\alpha_1, \ldots, \alpha_{k+e}$ be fixed distinct elements in $\mathbb{F}_p$. For specificity, we can pick the first $k+e$ elements. For any vector $\mathbf{u} = [u_1, \ldots, u_k]$ 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:

$$E_p(\mathbf{u}) = [f_\mathbf{u}(\alpha_1), \ldots, f_\mathbf{u}(\alpha_{k+e})].$$

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 $\mathbf{c} = \mathbf{c'}$.

Computing E_p amounts to evaluating a univariate polynomial of degree less than k with given coefficients in the $k+e$ points $\alpha_1, \ldots, \alpha_{k+e}$ over $\mathbb{F}_p$. The standard method to do so takes a number of arithmetic operations over $\mathbb{F}_p$ that is polynomial in $k+e$, and each arithmetic operation can be performed in time polynomial in $\log(p)$ using modular arithmetic. □

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.

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 $\mathcal{G}$ that works for every order and every polynomial.

During the construction, it will help us to have an upper bound on the size of the variable set X of the polynomial f, say $X = X_n \doteq \{x_1, \ldots, x_n\}$, and on the individual degree of f, say d. For a prime p to be determined, 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 7 provided $p \geq 2n$. 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 $\mathcal{G}_{n,d}$ to perform the following substitution on each variable x_i for $i \in [n]$:

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

Notice that if $\overline{m}$ is a monomial in X_n, then $\mathcal{G}_{n,d}(\overline{m})$ is a monomial in Y_{2n}. The erasure capability of E_p allows us to retrieve $\overline{m}$ from the restriction of $\mathcal{G}_{n,d}(\overline{m})$ to any half of the variables Y_{2n}.

Lemma 8. *Let n, d be positive integers, and $p \geq \max(2n, d+1)$ a prime. Let $Y_{2n} \doteq U \sqcup V$ 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_n. If $\overline{m}_1 \neq \overline{m}_2$, then $\mathcal{G}_{n,d}(\overline{m}_1)_U \neq \mathcal{G}_{n,d}(\overline{m}_2)_U$ and $\mathcal{G}_{n,d}(\overline{m}_1)_V \neq \mathcal{G}_{n,d}(\overline{m}_2)_V$.*

Proof. The condition that $p \geq 2n$ ensures that the gadget $\mathcal{G}_{n,d}$ is well-defined.

Let $\overline{m}$ be a monomial in X_n 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

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

Armed with this fact, let $\overline{m}_1, \overline{m}_2$ be monomials of individual degree at most d in X_n. Suppose $\overline{m}_1$, $\overline{m}_2$ have degree vectors $\mathbf{d}_1$, $\mathbf{d}_2$, respectively. Furthermore, suppose that for $U = \{y_{i_1}, \ldots, y_{i_n}\} \subseteq Y_{2n}$, we have that $\mathcal{G}_{n,d}(\overline{m}_1)_U = \mathcal{G}_{n,d}(\overline{m}_2)_U$. In other words, the powers of the U-variables in $\mathcal{G}_{n,d}(\overline{m}_1)$ and $\mathcal{G}_{n,d}(\overline{m}_2)$ 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 $p \geq d+1$ 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 $\mathcal{G}_{n,d}(\overline{m}_1)_U = \mathcal{G}_{n,d}(\overline{m}_2)_U$, then $\overline{m}_1 = \overline{m}_2$.

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

Now we show that our gadget $\mathcal{G}_{n,d}$ for an appropriate choice of p (depending on n and d) satisfies the conditions of Theorem 1. In fact, we prove a refined statement with Theorem 9 that considers the total degree $\deg(f)$ and the upper bound d on the individual degree. We denote the resulting gadget by $\mathcal{G}_{\text{ec}}$, where "ec" refers to "erasure code."

Theorem 9 (refined main result). *There exists an explicit oblivious gadget $\mathcal{G} = \mathcal{G}_{\mathrm{ec}}$ such that $\mathcal{G}_{n,d}$ maps an n-variate polynomial f with individual degree at most d and sparsity s to a $(2n)$-variate polynomial $\mathcal{G}_{n,d}(f)$ with individual degree $O(\deg(f)(n+d))$, sparsity s, and* $\mathsf{roABP\text{-}width}(\mathcal{G}_{n,d}(f)) = s$.

Theorem 1 follows from Theorem 9 by setting $d = \deg(f)$ in Theorem 9, which is a valid choice as the individual degree is always upper bounded by the total degree, and the fact that the total degree is at most n times the individual degree.

Proof of Theorem 9. Let $q \doteq \max(2n, d+1)$. By Bertrand's postulate, there always exists a prime $p \in [q, 2q]$. For specificity, we pick p to be the first prime in that range. We show that the family $\mathcal{G}_{n,d}$ defined by (1) satisfies the conditions of the theorem.

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

Now we establish the lower bound on roABP width for g. Let $\pi : [2n] \to [2n]$ be any variable order on Y_{2n}. 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

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

By Lemma 8, all monomials in the set $\{\mathcal{G}_{n,d}(\overline{m}_1)_U, \ldots, \mathcal{G}_{n,d}(\overline{m}_s)_U\}$ are distinct, as are all the ones in $\{\mathcal{G}_{n,d}(\overline{m}_1)_V, \ldots, \mathcal{G}_{n,d}(\overline{m}_s)_V\}$. Thus, for $i \in [s]$, the entry $(\mathcal{G}_{n,d}(\overline{m}_i)_U, \mathcal{G}_{n,d}(\overline{m}_i)_V)$ 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, that $\mathsf{roABP\text{-}width}_{\pi}(g) \geq s$. Since π was arbitrary, we have that $\mathsf{roABP\text{-}width}(g) \geq s$.

Since $\mathcal{G}_{n,d}$ maps distinct monomials of individual degree at most d in X_n to distinct monomials, g still has sparsity s, so there exists an roABP of width s computing g, no matter the variable order. Therefore, $\mathsf{roABP\text{-}width}(g) = s$.

Finally, note that $\mathcal{G}_{n,d}$ does not depend on the underlying field. The prime p can be determined in time polynomial in n and d by exhaustive search. Given p, the mapping E_p can be computed in time polynomial in n and $\log(p)$. It follows that the degree vector of $\mathcal{G}_{n,d}(x_i)$ can be computed in time polynomial in n and d for each $i \in [n]$. □

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 [vK85].

Fact 10. *For all fields $\mathbb{F}$ and positive integers n and d, there exists an n-variate polynomial f_1 over $\mathbb{F}$ of individual degree d and sparsity 2^n that has a factor g_1 of sparsity d^n. Specifically,*

$$f_1 = \prod_{i=1}^{n} (x_i^d - 1) \text{ and } g_1 = \prod_{i=1}^{n} (1 + x_i + x_i^2 + \cdots + x_i^{d-1}).$$

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., [Vol15, Example 32]).

Fact 11. *Suppose $d = \mathrm{char}(\mathbb{F}) \neq 0$. For all positive integers n, there exists an n-variate polynomial f_2 over $\mathbb{F}$ of degree d and sparsity n that has a factor g_2 of sparsity $\binom{n+d-2}{d-1}$. Specifically,*

$$f_2 = x_1^d + \cdots + x_n^d \text{ and } g_2 = (x_1 + \cdots + x_n)^{d-1}.$$

Notice that by the multinomial theorem and the fact that d is prime, every monomial of degree up to $d-1$ occurs in g_2 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 (factor complexity). *Suppose that f is an n-variate polynomial over $\mathbb{F}$ of degree $\deg(f) \leq d$ and sparsity s that has a factor g of sparsity s'. For $\mathcal{G} = \mathcal{G}_{\mathrm{ec}}$, $F = \mathcal{G}_{n,d}(f)$ is a $(2n)$-variate polynomial over $\mathbb{F}$ of individual degree $O(\deg(f)(n+d))$ with $\mathsf{roABP\text{-}width}(F) = s$ that has a factor G with $\mathsf{roABP\text{-}width}(G) = s'$. In particular, for every field $\mathbb{F}$ and positive integer n:*

1. *For every positive integer d, there exists a $(2n)$-variate polynomial F_1 over $\mathbb{F}$ of individual degree $O(dn(n+d))$ with $\mathsf{roABP\text{-}width}(F_1) = 2^n$ that has a factor G_1 with $\mathsf{roABP\text{-}width}(G_1) = d^n$.*

2. *If $\mathrm{char}(\mathbb{F}) = d \neq 0$, then there exists a $(2n)$-variate polynomial F_2 over $\mathbb{F}$ of individual degree $O(d(n+d))$ with $\mathsf{roABP\text{-}width}(F_2) = n$ that has a factor G_2 with $\mathsf{roABP\text{-}width}(G_2) = \binom{n+d-2}{d-1}$.*

Proof. Since $\mathcal{G}_{n,d}$ is a substitution, if g is a factor of f, then $G \doteq \mathcal{G}_{n,d}(g)$ is a factor of $F \doteq \mathcal{G}_{n,d}(f)$. Applying Theorem 1 to f and to g then yields the general result. The particular statements follow from instantiations based on Fact 10 (where $\deg(f_1) = dn$) and Fact 11, respectively. Plugging in the expressions for the sparsity yields $\mathsf{roABP\text{-}width}(F_1) = 2^n$ and $\mathsf{roABP\text{-}width}(G_1) = d^n$, and $\mathsf{roABP\text{-}width}(F_2) = n$ and $\mathsf{roABP\text{-}width}(G_2) = \binom{n+d-2}{d-1}$, respectively. □

Note that for fixed d, the width gap in Item 1 of Corollary 12 goes from $w = 2^n$ to $d^n = w^{\log d}$, and in Item 2 from $w = n$ to $\binom{n+d-2}{d-1} = w^{\Omega(d)}$. By setting $d = n$, we obtain an explicit family of polynomials with a quasi-polynomial gap $w \mapsto w^{\log w}$ in the case of Item 1, and an exponential gap $w \mapsto \binom{2w-2}{w-1} = \Theta(2^{2w}/\sqrt{w})$ in the case of Item 2. Note that the example of an exponential gap requires the underlying field to depend on w (and do so in a particular way), whereas the quasi-polynomial gap works over every fixed field.

Negative answers to the factor closure question are stronger when they exhibit an *irreducible* factor of high complexity rather than merely a factor. The following well-known technique of adding an auxiliary variable allows transforming gaps for arbitrary factors to gaps for irreducible factors up to some loss (see, e.g., [vK85, Example 5.1] for an earlier use). We include a proof for completeness.

Proposition 13. *Let $\mathbb{F}$ be a field and n a positive integer. For all nonzero n-variate polynomials f, g, h over $\mathbb{F}$ such that $f = gh$, there exist $(n+1)$-variate polynomials $\hat{f}, \hat{g}$ over $\mathbb{F}$ such that $\hat{g}$ is an irreducible factor of $\hat{f}$, $\mathsf{roABP\text{-}width}(f) \leq \mathsf{roABP\text{-}width}(\hat{f}) \leq \mathsf{roABP\text{-}width}(f) + \mathsf{roABP\text{-}width}(h)$, and $\mathsf{roABP\text{-}width}(g) \leq \mathsf{roABP\text{-}width}(\hat{g}) \leq \mathsf{roABP\text{-}width}(g) + 1$.*

Proof. Let z be a fresh variable. Define $\hat{g} = g + z$ and $\hat{f} = f + zh$. We have that $\hat{f} = gh + zh = \hat{g}h$, so $\hat{g}$ is a factor of $\hat{f}$. To see that $\hat{g}$ is irreducible, notice that any factorization of $\hat{g}$ must have a factor with z-degree 1 and a factor with z-degree 0. Furthermore, the factor with z-degree 0 must divide the coefficient of z in $\hat{g}$, which is 1, so said factor must be a constant.

As setting the variable z to zero cannot increase roABP width, we have $\mathsf{roABP\text{-}width}(f) \leq \mathsf{roABP\text{-}width}(\hat{f})$ and $\mathsf{roABP\text{-}width}(g) \leq \mathsf{roABP\text{-}width}(\hat{g})$. The remaining inequalities follow from the subadditivity of $\mathsf{roABP\text{-}width}$, and the facts that $\mathsf{roABP\text{-}width}(z) = 1$ and $\mathsf{roABP\text{-}width}(hz) = \max(\mathsf{roABP\text{-}width}(h), \mathsf{roABP\text{-}width}(z)) = \mathsf{roABP\text{-}width}(h)$ as z is not a variable of h and h is nonzero. □

Proposition 13 allows us to derive the following result for irreducible factors from the two instantiations in Corollary 12.

Corollary 14 (irreducible factor complexity). *For every field $\mathbb{F}$ and every positive integer n:*

1. *For every positive integer d, there exists a $(2n+1)$-variate polynomial $\hat{F}_1$ over $\mathbb{F}$ of individual degree $O(dn(n+d))$ with $2^n \leq \mathsf{roABP\text{-}width}(\hat{F}_1) \leq 2^{n+1}$ that has an irreducible factor $\hat{G}_1$ with $d^n \leq \mathsf{roABP\text{-}width}(\hat{G}_1) \leq d^n + 1$.*

2. *If $\mathrm{char}(\mathbb{F}) = d \neq 0$, then there exists a $(2n+1)$-variate polynomial $\hat{F}_2$ over $\mathbb{F}$ of individual degree $O(d(n+d))$ with $n \leq \mathsf{roABP\text{-}width}(\hat{F}_2) \leq 2n$ that has an irreducible factor $\hat{G}_2$ with $\binom{n+d-2}{d-1} \leq \mathsf{roABP\text{-}width}(\hat{G}_2) \leq \binom{n+d-2}{d-1} + 1$.*

Proof. We apply $\mathcal{G} = \mathcal{G}_{\mathrm{ec}}$ as in the proof of Corollary 12. In Fact 10, we can see that $h_1 \doteq f_1/g_1 = \prod_{i=1}^{n}(x_i - 1)$ has sparsity 2^n, so the $\mathsf{roABP\text{-}width}$ of $H_1 \doteq \mathcal{G}_{n,d}(h_1) = \mathcal{G}_{n,d}(f_1)/\mathcal{G}_{n,d}(g_2) = F_1/G_1$ is also 2^n. Applying Proposition 13, we obtain Item 1. Item 2 follows in a similar way but using Fact 11, where $h_2 \doteq f_2/g_2 = x_1 + \cdots + x_n$ has sparsity n. □

Similar to before, by setting $d = n$, we obtain an explicit family of polynomials with a quasi-polynomial gap in the case of Item 1, namely from w to at least $w^{\Omega(\log w)}$ over every field. In the case of Item 2, the gap is exponential, namely from $w \in [n, 2n]$ to at least $\binom{2n-2}{n-1} \geq \binom{w-2}{w/2-1} = \Theta(2^w/\sqrt{w})$ under the aforementioned restrictions on the underlying fields.

4 Prior results

In this section, we present the prior gadgets that were used for lifting sparsity gaps to gaps in roABP width. Our gadget does not rely on them, but the exposition enables a technical comparison, which we present at the end of this section.

Restricted variable orders. The following gadget was used for lifting sparsity to roABP width with respect to restricted variable orders in various contexts [FSTW21, HLT24, AAD+26]. In our presentation, the restriction is that the odd-indexed variables are read before the even-indexed variables. An equivalent statement is obtained by instead requiring the first half of the variables to be read before the second half. Our view has the advantage that the gadget becomes independent of the number of variables (in addition to being independent of the individual degree).

Theorem 15 (gadget for restricted variable orders). *There exists an explicit oblivious gadget $\mathcal{G}_{\mathrm{bip}}$ that maps an n-variate polynomial f of degree d and sparsity s to a $(2n)$-variate polynomial $\mathcal{G}_{\mathrm{bip}}(f)$ that has the same individual degree as f, degree $2d$, and $\mathsf{roABP\text{-}width}_\pi(\mathcal{G}_{\mathrm{bip}}(f)) = s$ for every variable order $\pi : [2n] \to [2n]$ that puts the variables with odd index before the ones with even index.*

Proof. We define

$$\mathcal{G}_{\mathrm{bip}}(x_i) = y_{2i-1}y_{2i} \tag{2}$$

for each $i \in [n]$, and write $g = \mathcal{G}_{\mathrm{bip}}(f)$. The individual degree of y_{2i-1} and y_{2i} 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 $Y_{2n} = \{y_1, \ldots, y_{2n}\}$ into $U = \{y_1, y_3, \ldots, y_{2n-1}\}$ and $V = \{y_2, \ldots, y_{2n}\}$. Notice that for any monomial $\overline{m}$ appearing in g, $\overline{m}_U$ uniquely determines $\overline{m}_V$ and vice versa, since $\overline{m}_U$ and $\overline{m}_V$ have the same degree vectors (when viewed as vectors of length $|U| = |V|$). Therefore, $\mathrm{CMat}_{U,V}(g)$ is a matrix with at most one nonzero entry in each row and column, and the number of nonzero entries is s. So, $\mathrm{rank}(\mathrm{CMat}_{U,V}(g)) = s$ and Lemma 6 implies $\mathsf{roABP\text{-}width}_\pi(g) \geq s$ as long as the permutation π keeps the variables in U before the variables in V. As $\mathcal{G}_{\mathrm{bip}}$ maps distinct monomials to distinct monomials, the sparsity of $\mathcal{G}_{\mathrm{bip}}(f)$ is the same as the sparsity of f. It follows that the inequality $\mathsf{roABP\text{-}width}_\pi(g) \leq s$ also holds.

$\mathcal{G}_{\mathrm{bip}}$ does not depend on the underlying field. The degree vector of $\mathcal{G}_{\mathrm{bip}}(x_i)$ can be computed in time polynomial in n as it is identically zero except for ones in positions $2i-1$ and $2i$. □

The subscript "bip" in our notation $\mathcal{G}_{\mathrm{bip}}$ for the gadget is a shorthand for "bipartite." The choice relates to the view of a quadratic multilinear gadget like $\mathcal{G}_{\mathrm{bip}}$ as a graph with edges corresponding to the original variables X_n and vertices to the new variables Y_{2n}. In the case of $\mathcal{G}_{\mathrm{bip}}$, the underlying graph is bipartite, namely with bipartition $U \sqcup V$, where U consists of the odd-indexed variables in Y_{2n} and V of the even-indexed ones.

Unrestricted variable orders. In order to lift the sparsity gap result of Fact 10 to a gap result for roABP with respect to *every* variable order, [AAD+26] adapt the gadget $\mathcal{G}_{\mathrm{bip}}$ into the gadget $\mathcal{G}_{\mathrm{exp}}$ by replacing the underlying graph by a certain (non-bipartite) graph G_n. The graph $G_n = (Z_\ell, X_n)$ has n edges corresponding to the original variables X_n, and ℓ vertices corresponding to the new variables, which we now denote as $Z_\ell = \{z_1, \ldots, z_\ell\}$ to avoid ambiguity in the further analysis. For each $i \in [n]$,

$$\mathcal{G}_{\mathrm{exp}}(x_i) = z_{u_i}z_{v_i}, \tag{3}$$

where $e_i = (z_{u_i}, z_{v_i})$ is the edge of G_n corresponding to x_i. The graph G_n is chosen to be sparse and well-connected. More specifically, G_n is a graph of degree at most some constant Δ with the following connectivity property:

$$\begin{aligned}&\text{For every pair } S, T \text{ of disjoint subsets of vertices with } \min(|S|, |T|) \geq |Z_\ell|/3,\\ &G_n = (Z_\ell, X_n) \text{ has at least one edge } (u, v) \in S \times T.\end{aligned} \tag{4}$$

Such graphs G_n can be constructed explicitly for every positive integer n from certain expander families (see Proposition 19 in the appendix), which is what "exp" in the subscript of $\mathcal{G}_{\mathrm{exp}}$ refers to.

The defining properties of G_n imply the existence of a large induced matching between the parts of any *balanced* bipartition of the vertices, i.e., a partition into two subsets where the two subsets differ by at most one in size.

Claim 16. *For every balanced partition $U \sqcup V$ of the vertices of $G_n = (Z_\ell, X_n)$, there exists an induced subgraph of G_n that is a matching $M \subseteq U \times V$ of size $|M| \geq \gamma \cdot |X_n|$, where $\gamma = \frac{1}{6\Delta^2}$.*

Proof. We construct M greedily. Start with the empty matching M, $S = U$, and $T = V$. As long as S and T meet the size lower bound for (4), iteratively add the edge (u, v) to M and remove the vertices u and v as well as their neighbors from S and T. The process has as an invariant that $M \subseteq U \times V$ is a matching that can be obtained as an induced subgraph of G. The choice of $(u, v) \in S \times T$ guarantees that $M \subseteq U \times V$. The removal of u and v guarantees that M is a matching. The choice of (u, v) as an edge of G_n and the removal of the neighbors of u and v guarantee that M can be obtained as an induced subgraph of G_n.

Each iteration of the process adds one edge to M and removes at most 2Δ vertices. By the balancedness of the partition $U \sqcup V = Z_\ell$, $\min(|S|, |T|) \geq \frac{\ell-1}{2}$ at the start of the process. At the start of the kth iteration, $\min(|S|, |T|) \geq \frac{\ell-1}{2} - 2(k-1)\Delta$, which meets the lower bound of $\ell/3$ for (4) as long as $k \leq \frac{\ell}{12\Delta}$. Thus, at the end of the process, $|M| \geq \frac{\ell}{12\Delta}$. As $G_n = (Z_\ell, X_n)$ has degree at most Δ, we have that $n = |X_n| \leq \frac{\Delta}{2} \cdot |Z_\ell| = \frac{\Delta}{2} \cdot \ell$, so $\frac{\ell}{12\Delta} \geq \frac{n}{6\Delta^2}$ and the conclusion follows. □

Consider applying the resulting gadget $\mathcal{G}_{\exp}$ to an n-variate polynomial f. For any given balanced partition $U \sqcup V$ of the variables Z_ℓ of $\mathcal{G}_{\exp}(f)$, let M denote an induced matching as in Claim 16. Let X' denote the original variables $x_i \in X_n$ such that $e_i \in M$, and Z' the variables $z_j \in Z_\ell$ that appear as a vertex of an edge in M. For each $x_i \in X'$, set $y_{2i-1} = z_{u_i}$ and $y_{2i} = z_{v_i}$. The fact that M is a matching guarantees that these settings are compatible. The fact that M is an induced subgraph of G guarantees that setting the variables z_j outside of Z' in $\mathcal{G}_{\exp}(f)$ to zero yields the same polynomial as $\mathcal{G}_{\text{bip}}(f')$, where f' denotes f with the variables outside of X' set to zero. In symbols:

$$\mathcal{G}_{\exp}(f)|_{Z_\ell \setminus Z' \leftarrow 0} = \mathcal{G}_{\text{bip}}(f|_{X_n \setminus X' \leftarrow 0}).$$

One can then apply Theorem 15 to $\mathcal{G}_{\text{bip}}(f')$ and obtain a lower bound of the sparsity of f' for

$$\mathsf{roABP\text{-}width}_\pi(\mathcal{G}_{\text{bip}}(f')) = \mathsf{roABP\text{-}width}_\pi(\mathcal{G}_{\exp}(f)|_{Z_\ell \setminus Z' \leftarrow 0}) \leq \mathsf{roABP\text{-}width}_\pi(\mathcal{G}_{\exp}(f)),$$

where π is any order that puts the variables z_j in U before those in V. Taking the minimum over all possible choices of U and V shows that $\mathsf{roABP\text{-}width}(\mathcal{G}_{\exp}(f))$ is at least the minimum sparsity of the zero substitutions f' of f that leave at least $\gamma \cdot n$ of the original variables X_n free.

Modulo explicitness, we have established the following generalization of the [AAD+26] approach. Note that the constructed gadget $\mathcal{G}_{\exp}$ requires an upper bound on the number of variables (because of the graph G_n) but no upper bound on the (individual) degree of the polynomial.

Theorem 17 (prior gadget for all variable orders). *For some positive constant γ, there exists an explicit oblivious gadget $\mathcal{G} = \mathcal{G}_{\exp}$ such that $\mathcal{G}_n$ maps an n-variate polynomial f of degree d and sparsity s to an $O(n)$-variate polynomial $\mathcal{G}_n(f)$ of degree $2d$, sparsity s, and $\mathsf{roABP\text{-}width}(\mathcal{G}_n(f)) \geq s'$, where s' is the minimum sparsity of any polynomial f' obtained from f by setting all but at least $\gamma \cdot n$ of the n variables to zero.*

For completeness and concreteness, we describe a construction that realizes the required explicitness of the gadget $\mathcal{G}_{\exp}$ in Proposition 19 in the appendix.

Theorem 17 has the following immediate consequence for factor complexity, similar to Corollary 12 in the context of our gadget.

Corollary 18. *Suppose that f is an n-variate polynomial over $\mathbb{F}$ of degree d and sparsity s that has a factor g with the following property: Every polynomial g' obtained from g by setting all but at least $\gamma \cdot n$ of the n variables to zero has sparsity at least s'. For $\mathcal{G} = \mathcal{G}_{\exp}$, $F = \mathcal{G}_n(f)$ is an $O(n)$-variate polynomial over $\mathbb{F}$ of degree $2d$ with* $\mathsf{roABP\text{-}width}(F) \leq s$ *that has a factor G with* $\mathsf{roABP\text{-}width}(G) \geq s'$.

[AAD+26] instantiates Corollary 18 over any field $\mathbb{F}$ with the polynomials f_1 given in Fact 10, for which $s = 2^n$ and $s' = d^{\gamma \cdot n}$. Over fields $\mathbb{F}$ with characteristic $d \neq 0$, Corollary 18 can equally be instantiated with the polynomials f_2 from Fact 11, for which $s = n$ and $s' = \binom{\gamma \cdot n + d - 2}{d-1}$. Similar to the corresponding applications of our gadget, setting $d = \Theta(n)$ yields explicit families of polynomials with a gap that is at least quasi-polynomial over every fixed field and exponential with restrictions on the underlying fields. The gaps obtained from applying the prior gadget $\mathcal{G}_{\exp}$ are somewhat smaller than the ones that follow from our gadget $\mathcal{G}_{\mathrm{ec}}$. In the case of f_1, setting $d = n$ brings us from width w to width at least $w^{\gamma \log w}$ instead of $w^{\log w}$. In the case of f_2, we go from width w to width at least $\Theta(2^{(1+\gamma)w}/\sqrt{w})$ (setting $d = \gamma \cdot n$) instead of width $\Theta(2^{2w}/\sqrt{w})$ (setting $d = n$). Note also that the analysis of the weight gap only gives lower bounds for $\mathcal{G}_{\exp}$ whereas it is exact for $\mathcal{G}_{\mathrm{ec}}$.

The similarity extends to *irreducible* factor complexity by applying Proposition 13. The translation induces some minor quantitative losses, but the qualitative comparison between the results obtained using $\mathcal{G}_{\exp}$ versus $\mathcal{G}_{\mathrm{ec}}$ remains.

Conclusion. In comparison to the approach in [AAD+26], our technique has the advantage of being always applicable, exactly preserving the complexity measure, and being entirely elementary. Obtaining strong results along the lines of [AAD+26] requires that the sparsity of the polynomial does not reduce significantly under zero-substitutions. Even with this structural property, the technique still weakens the lower bound and becomes non-tight because the argument of n gets reduced to $\gamma \cdot n$. Our approach translates every polynomial f into a polynomial f' such that the roABP width of f' is always exactly equal to the sparsity of f. Such tight control helps in applications like factor complexity. In terms of derivation, the construction in [AAD+26] hinges on the existence of certain expander families, which, though a common tool, are nontrivial to develop from scratch. In contrast, modulo the roABP width characterization (Lemma 6), which [AAD+26] needs as well, this paper contains a complete derivation from first principles, including the required background on error-correcting codes (Lemma 7). On the other hand, [AAD+26] has the advantage that the degree of the polynomial increases by a constant factor, while our gadget increases the degree by a factor of $O(n(n+d))$.

5 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 [AAD+26]. 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 established lower bounds for specific instances of roABP-IPS [FSTW21, HLT24, HLT26]. 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 [AAD+26], 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. [BSV20] 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 roABP width? Alternatively, can one find larger gaps in roABPs width independent of any results on sparsity?

Acknowledgements

The authors thank Nutan Limaye for useful discussions, in particular for background on algebraic proof systems, and anonymous reviewers for helpful suggestions. This material is based upon work supported by the National Science Foundation under Grant Nos. 2137424 and 2312540.

Appendix: Explicitness of the prior gadget

In this section, we show how certain explicit families of expanders can be used to construct the graphs G_n for every positive integer n in time polynomial in n. Those graphs are needed for the prior gadget $\mathcal{G}_{\text{exp}}$ but not for our gadget $\mathcal{G}_{\text{ec}}$. The properties required of the expander family are that the expanders can be constructed in time polynomial in number of vertices and that the gap between vertex set sizes is linearly bounded. Those requirements can be achieved. We refer to [HLW06] for the required background on expander graphs.

Proposition 19. *There exists a family of graphs G_n where n ranges over all positive integers such that G_n contains exactly n edges, has degree bounded by some constant Δ, satisfies* (4)*, and can be constructed deterministically in time polynomial in n.*

Proof. The starting point is an explicit family of expander multigraphs with at most a linear gap between the numbers of vertices of successive graphs in the family: If the numbers are $\ell_1 < \ell_2 < \ldots$, then $\ell_{i+1} \leq c \cdot \ell_i$ for some constant c. An example of such a family consists of the graphs C_p for primes p, where C_p denotes the p-cycle with inverse chords; the vertex set of C_p is $\mathbb{Z}_p$, and each vertex $v \in \mathbb{Z}_p$ has three edges: one to $v-1$, one to $v+1$, and one to the multiplicative inverse v^{-1}, where all arithmetic is modulo p (and the multiplicative inverse of 0 is interpreted as 0). By Bertrand's postulate, this family satisfies the gap requirement with $c = 2$. The family is regular of degree 3, and is known to have a normalized spectral gap δ at least some positive constant. By taking a sufficiently high constant power (in terms of the adjacency matrix), we obtain an explicit family of multigraphs with the same vertex numbers $p_1 < p_2 < \ldots$, regular of degree some constant Δ', and with normalized spectral gap more than $2/3$. The latter level of spectral expansion implies that the ith multigraph in the family meets the connectivity requirements in (4) with $\ell = p_i$. This follows from the expander mixing lemma, which states that for every Δ'-regular multigraph (V, E)

and every $S, T \subseteq V$,

$$\left| \frac{|S \times T \cap E|}{\Delta' \cdot |V|} - \frac{|S| \cdot |T|}{|V|^2} \right| \leq \lambda \cdot \frac{\sqrt{|S| \cdot |T|}}{|V|},$$

where $\lambda \doteq 1 - \delta$. The lemma implies that for any two subsets $S, T \subseteq V$ with $|S| \cdot |T| > \lambda^2 \cdot |V|^2$, the multigraph contains at least one edge of the form $(u, v) \in S \times T$. This gives us the desired connectivity property for $\lambda < 1/3$, or equivalently, for $\delta > 2/3$.

The family constructed thus far consists of multigraphs rather than graphs; there can be parallel edges as well as self-loops. However, we can drop those types of edges without affecting the connectivity property (4). This gives us a graph H_p for every prime p such that H_p has exactly p vertices, degree at most Δ' and the desired connectivity property. The only remaining issue is that the family does not contain a graph with the prescribed number of edges n for *every* positive integer n.

Recall that our notation defines G_n to be a graph with n *edges*, and H_p as a graph with p *vertices*. To construct G_n, first find the largest prime p such that $p \cdot \Delta'/2 \leq n$. The inequality guarantees that H_p does not contain more than n edges, say $n' \leq n$. The largeness condition means that $p' \cdot \Delta'/2 > n$, where p' is the next prime after p. Since $p' \leq 2p$, this implies that $p \geq n/\Delta'$. In turn, this allows us to augment H_p with $n - n' \leq n$ new edges between existing vertices without increasing the degree of any vertex by more than $\frac{2(n-n')}{p} \leq \frac{2n}{p} \leq \frac{2n\Delta'}{n} = 2\Delta'$. This gives us the graph G_n with exactly n edges, degree at most $\Delta = 3\Delta'$, and having the connectivity property (4).

Algorithmically, starting from H_p, we iteratively find two vertices, say the lexicographically first two, whose degrees are still less than Δ, and add an edge between them. The above analysis shows that we can successfully execute $n - n'$ iterations. The overall process runs in time polynomial in n, as does the determination of p and the construction of H_p. Thus, the graph G_n can be constructed in time polynomial in n. □

References

[AAD+26] 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.

[AFS+18] 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.

[AGKS15] 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.

[BKR+25] 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.

[BS21] Pranav Bisht and Nitin Saxena. Blackbox identity testing for sum of special ROABPs and its border class. *Comput. Complex.*, 30(1):8, 2021.

[BSV20] 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.

[DMM24] 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.

[dRGR22] Susanna F. de Rezende, Mika Göös, and Robert Robere. Guest column: Proofs, circuits, and communication. *SIGACT News*, 53(1):59–82, 2022.

[dRMN+20] 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.

[dRV25] 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.

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

[FS13] 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.

[FSS14] 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.

[FSTW21] 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.

[GG20] 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.

[GGKS20] Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. *Theory Comput.*, 16:1–30, 2020.

[GKST17] 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.

[GPW18] Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. *SIAM J. Comput.*, 47(6):2435–2450, 2018.

[GPW20] Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. *SIAM J. Comput.*, 49(4), 2020.

[HLT24] 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.

[HLT26] Tuomas Hakoniemi, Nutan Limaye, and Iddo Tzameret. Hard CNF instances for ideal proof systems. *CoRR*, abs/2605.04544, 2026.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the AMS*, 43(4):439–561, 2006.

[Kal89] Erich L. Kaltofen. Factorization of polynomials given by straight-line programs. *Adv. Comput. Res.*, 5:375–412, 1989.

[KNS20] 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.

[LMM+22] 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.

[Nis91] 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.

[RM99] Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Comb.*, 19(3):403–435, 1999.

[RS60] 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.

[RS05] Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Comput. Complex.*, 14(1):1–19, 2005.

[Sok20] Dmitry Sokolov. (Semi)Algebraic proofs over $\{\pm 1\}$ variables. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, page 78–90. Association for Computing Machinery, 2020.

[ST21] Amit Sinhababu and Thomas Thierauf. Factorization of polynomials given by arithmetic branching programs. *Computational Complexity*, 30(2):15:1–15:47, 2021.

[vK85] Joachim von zur Gathen and Erich L. Kaltofen. Factoring sparse multivariate polynomials. *J. Comput. Syst. Sci.*, 31(2):265–287, 1985.

[Vol15] Ilya Volkovich. Deterministically factoring sparse polynomials into multilinear factors and sums of univariate polynomials. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, Princeton, NJ, USA, August 24-26, 2015*, volume 40 of *LIPIcs*, pages 943–958. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il