NP-Hardness of Testing Equivalence to Sparse Polynomials and to Constant-Support Polynomials

Omkar Baraskar
University of Waterloo, Canada

Agrim Dewan
Indian Institute of Science, Bengaluru, India

Chandan Saha
Indian Institute of Science, Bengaluru, India

Pulkit Sinha
University of Waterloo, Canada

Abstract

An s-sparse polynomial has at most s monomials with nonzero coefficients. The Equivalence Testing problem for sparse polynomials (ETsparse) asks to decide if a given polynomial f is equivalent to (i.e., in the orbit of) some s-sparse polynomial. In other words, given $f \in \mathbb{F}[\mathbf{x}]$ and $s \in \mathbb{N}$, ETsparse asks to check if there exist $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$ and $\mathbf{b} \in \mathbb{F}^{|\mathbf{x}|}$ such that $f(A\mathbf{x} + \mathbf{b})$ is s-sparse. We show that ETsparse is NP-hard over any field $\mathbb{F}$, if f is given in the sparse representation, i.e., as a list of nonzero coefficients and exponent vectors. This answers a question posed by Gupta, Saha and Thankey (SODA 2023) and also, more explicitly, by Baraskar, Dewan and Saha (STACS 2024). The result implies that the Minimum Circuit Size Problem (MCSP) is NP-hard for a *dense* subclass of depth-3 arithmetic circuits if the input is given in sparse representation. We also show that approximating the smallest s_0 such that a given s-sparse polynomial f is in the orbit of some s_0-sparse polynomial to within a factor of $s^{\frac{1}{3}-\epsilon}$ is NP-hard for any $\epsilon > 0$; observe that s-factor approximation is trivial as the input is s-sparse. Finally, we show that for any constant $\sigma \geq 6$, checking if a polynomial (given in sparse representation) is in the orbit of some support-σ polynomial is NP-hard. Support of a polynomial f is the maximum number of variables present in any monomial of f. These results are obtained via direct reductions from the 3-SAT problem.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Equivalence testing, MCSP, sparse polynomials, 3SAT

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.16

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2024/104

Funding *Chandan Saha*: Partially supported by a MATRICS grant of the Science and Engineering Research Board, DST, India.
Pulkit Sinha: Partially supported by a Mike and Ophelia Lazardis Fellowship, and partially supported by NSERC Canada.

Acknowledgements We thank the anonymous reviewers for their detailed and constructive feedback, which has helped us improve the presentation of this work. In particular, we thank one of the reviewers for pointing out some inaccuracies in the original proofs of Lemmas 50 and 51; simpler proofs for both lemmas came up in the process of fixing these inaccuracies.

1 Introduction

The Polynomial Equivalence (PE) problem asks to decide if two polynomials, given as lists of coefficients, are equivalent. Polynomials $f, g \in \mathbb{F}[\mathbf{x}]$ are *equivalent*, denoted as $f \sim g$, if there is an $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$ and a $\mathbf{b} \in \mathbb{F}^{|\mathbf{x}|}$ such that $f = g(A\mathbf{x} + \mathbf{b})$. Equivalent polynomials

51st International Colloquium on Automata, Languages, and Programming (ICALP 2024).
Editors: Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson;
Article No. 16; pp. 16:1–16:21
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

represent the same function up to a change of the coordinate system.[1] The PE problem is thus regarded as the algebraic analog of the graph isomorphism (GI) problem. PE is at least as hard as GI [2, 36], but we do not know if it is much harder than GI. There is, in fact, a cryptographic authentication scheme based on the presumed average-case hardness of PE [46]. Is PE NP-hard? Over finite fields, PE is not NP-hard unless the polynomial hierarchy collapses [51, 55]. In contrast, PE is not even known to be decidable over $\mathbb{Q}$. With the aim of gaining more insight into the complexity of testing polynomial equivalence, a natural variant of PE has been studied in the literature. This variant is known as *equivalence testing*.

In the following discussion, whenever we write "circuit(s)" and "formula(s)", we mean arithmetic circuit(s) and arithmetic formula(s), respectively, unless mentioned otherwise.[2]

Equivalence testing. Equivalence testing (ET) comes in two flavors – ET for polynomial families and ET for circuit classes. ET for a polynomial family $\mathcal{F}$ is defined as follows: given a *single* polynomial f, check if it is equivalent to some $g \in \mathcal{F}$. This variant of PE was introduced in [37, 36], wherein randomized polynomial-time ET algorithms were provided for the permanent, determinant, and elementary and power symmetric polynomial families. Subsequently, efficient ET algorithms were given for various other important polynomial families, such as the iterated matrix multiplication (IMM) family [39] (see the Related Works section in the full version). These algorithms are efficient even if f is provided as a circuit or a black-box.[3] ET for a circuit class $\mathcal{C}$ (a.k.a testing equivalence to $\mathcal{C}$) is defined similarly: given a polynomial f, decide if it is equivalent to some polynomial g that is computable by a circuit in $\mathcal{C}$. Recently, efficient ET algorithms have been given for read-once formulas [24] and a special subclass of sparse polynomials, namely t-design polynomials for constant t [8]. Sparse polynomials are depth-2 circuits.[4] It is natural to ask whether or not ET can be solved efficiently for *general* sparse polynomials. This question was posed in [24] and also, more explicitly, in [8].

Before proceeding to discuss ET for sparse polynomials, we point out a subtle difference between ET for polynomial families and that for circuit classes. The polynomial families for which ET has been studied so far are such that if f is equivalent to some g in the family, then g is unique and it can be readily identified from f. For example, if f is equivalent to some determinant polynomial[5], then we know which one simply from the number of variables of f. Moreover, polynomials in most of these families admit well-known polynomial-size circuits. So, a circuit for g can be derived once it is identified. Thus, if f is also given as a circuit, then ET for such a family reduces to PE with the input polynomials given as circuits. Over finite fields, this version of PE is in $\mathsf{AM} \cap \mathsf{coAM}$ and hence unlikely to be NP-hard. On the other hand, in the case of ET for a circuit class, if f is equivalent to some circuit C in the class, then C need *not* be unique, and further, C may not be easily deducible from f. This leaves us with the prospect of proving that ET is hard for some natural circuit class. Do sparse polynomials form such a class?

[1] Over $\mathbb{R}$, an invertible map $\mathbf{x} \mapsto A\mathbf{x} + \mathbf{b}$ is simply a combination of rotation, reflection, scaling, and translation.

[2] An *arithmetic circuit* is like a Boolean circuit but with AND and OR replaced by $\times$ and $+$ gates, and with edges labelled by $\mathbb{F}$-elements. It computes a polynomial over $\mathbb{F}$. A *formula* is a circuit whose underlying graph is a tree.

[3] Black-box access to f means oracle access to f, we get $f(\mathbf{a})$ from a query point $\mathbf{a}$ in unit time. It is as if f is given as a "hidden" circuit, and the only operation we are allowed is to evaluate the circuit at chosen points.

[4] We assume that a depth-2 circuit has a $+$ gate on top and a bottom layer of $\times$ gates. If the top gate is a $\times$ gate, then ET can be solved efficiently using polynomial factorization algorithms [35].

[5] The n^2-variate determinant polynomial is the determinant of the matrix $(x_{i,j})_{i,j\in[n]}$ of formal variables.

ET for sparse polynomials. An n-variate, degree-d polynomial is *s-sparse* if it has at most s monomials with nonzero coefficients. An s-sparse polynomial is computable by a depth-2 circuit having top fan-in s. Sparse polynomials have been extensively studied in algebraic complexity, particularly with regard to identity testing [41, 43], interpolation [10, 21, 41, 12], and factorization [56, 11] (see the tutorial [49] and the references therein for more algorithms involving sparse polynomials). ET provides yet another avenue to understand these "basic" polynomials better. ET for sparse polynomials asks to check if a given polynomial is sparse in some coordinate system. More formally, given a polynomial f as an arithmetic circuit and an $s \in \mathbb{N}$, decide if there is an s-sparse polynomial g such that $f \sim g$. This problem was studied in [20] over $\mathbb{Q}$, wherein an exponential in n^4 time algorithm was provided. There has not been any significant progress on this problem since that work. The lack of improvements in the complexity for over three decades makes one wonder:

Is ET for sparse polynomials NP-hard?

In this work, we answer this question in the affirmative over *any* field (see the first part of Theorem 2) even if the input f is provided as a depth-2 circuit. The result answers the question posed in [24, 8]. To our knowledge, the theorem gives the first example of a natural circuit class for which ET is provably hard.

Although ET for sparse polynomials (ETsparse) is a fairly natural problem, there is a deeper reason to study ETsparse originating from the expressive power of affine projections of sparse polynomials and the *Minimum Circuit Size Problem* (MCSP) for depth-3 circuits. We discuss this reason below to motivate ETsparse when the input is a *homogeneous* polynomial.

1.1 ETsparse and MCSP for depth-3 circuits

First, we need a few definitions: A polynomial g is an *affine projection* of f if $g = f(A\mathbf{x} + \mathbf{b})$ for some $A \in \mathbb{F}^{|\mathbf{x}| \times |\mathbf{x}|}$ and $\mathbf{b} \in \mathbb{F}^{|\mathbf{x}|}$. If $\mathbf{b} = 0$, we say g is a *linear projection* of f; additionally, if $A \in \mathrm{GL}(|\mathbf{x}|)$, we say g is in the *orbit* of f, denoted as $\mathrm{orb}(f)$. Depth-3 circuits form a highly expressive class [23, 54]. A depth-3 ($\Sigma\Pi\Sigma$) circuit is a circuit with a + gate on top, a middle layer of × gates, and a bottom layer of + gates. A depth-3 circuit with a top fan-in of s is an affine projection of an s-sparse polynomial. Thus, the problem of deciding if a given f is an affine projection of an s-sparse polynomial is closely related to MCSP for depth-3 circuits. We say "closely related to" instead of "the same as" because the size of a depth-3 circuit is determined by not only its top fan-in but also its formal degree.

MCSP. The complexity of MCSP for Boolean circuits has baffled researchers for over six decades. MCSP for a Boolean circuit class $\mathcal{C}$ ($\mathcal{C}$-MCSP) takes input the truth table of an n-variate Boolean function f and a parameter $s \in \mathbb{N}$ and asks to check if f is computable by a circuit in $\mathcal{C}$ of size at most s. There are intriguing connections between MCSP and several other areas such as cryptography [34, 3], learning theory [14], average-case complexity [27], and proof complexity [47]. Whether or not MCSP for general Boolean circuits is NP-hard is a long-standing open question. It is known that MCSP is NP-hard for DNF [44, 4] and DNF $\circ$ XOR formulas [29]. But no NP-hardness result is known (under deterministic polynomial-time reductions) for more general circuit models such as AC^0 circuits.[6] This is not too surprising

[6] However, strong hardness results are known for several powerful circuit models under randomized or quasi-polynomial time or subexponential time reductions [30, 32, 31, 28].

as [34] showed that NP-hardness of $\mathcal{C}$-MCSP under *natural*[7] deterministic polynomial-time reductions implies a $2^{\Omega(n)}$ lower bound for $\mathcal{C}$, unless NP $\subseteq$ SUBEXP. Unfortunately, such strong lower bounds are not known even for depth-3 Boolean circuits. However, a $2^{\Omega(n)}$ lower bound is known for XOR ∘ AND ∘ XOR formulas [48], which are depth-3 arithmetic circuits over $\mathbb{F}_2$ and are like DNF ∘ XOR formulas but with the top OR gate replaced by an XOR gate. In fact, a $2^{\Omega(n)}$ lower bound is known for depth-3 arithmetic circuits over any fixed finite field [22]. This raises hope that we will be able to prove the hardness of MCSP for depth-3 arithmetic circuits over finite fields. But how is the input given in the case of MCSP for arithmetic circuits? And what about depth-3 circuits over fields of characteristic 0?

MCSP for arithmetic circuits: Input representation and model of computation. In the Boolean setting of MCSP, one of the main reasons to assume the input to be a truth table is that the assumption puts MCSP in NP. Analogously, in the algebraic setting, we could assume that the polynomial is given in the dense representation as a list of $\binom{n+d}{n}$ coefficients. But observe that even if the input is given as an arithmetic circuit, MCSP is in the complexity class MA over finite fields. This is because verifying if two circuits compute the same polynomial is the polynomial identity testing problem, which admits a randomized polynomial-time algorithm [16, 57, 52]. Furthermore, class MA equals NP, assuming a widely believed circuit lower bound [33]. A succinct input representation also opens up the possibility of proving NP-hardness of MCSP for models, such as depth-3 circuits over fields of characteristic 0, for which strong exponential lower bounds are unknown (the MCSP hardness to lower bound implication in [34] needs the input in the dense format). The current best lower bound for depth-3 circuits over fields of characteristic 0 is quasi-polynomial in n [42, 5].

It is, therefore, reasonable to assume that the input polynomial is given succinctly as a circuit which should only facilitate our efforts in proving NP-hardness of MCSP for arithmetic circuit classes. For example, there is an instance in the Boolean setting wherein succinct representation of the input helped prove NP-hardness of MCSP long before such a hardness result was shown with respect to the dense representation – it is the case of the *partial* MCSP problem [25, 28]. In this work, we assume that the input is given as a depth-2 circuit, i.e., as a list of nonzero coefficients, and exponent vectors in unary – this is the *sparse representation*.[8]

A few remarks are in order concerning the model of computation. Over finite fields, we assume the Turing machine model. However, over arbitrary fields of characteristic 0, it is natural to consider an arithmetic model of computation (similar to the Blum-Shub-Smale machine model [13]) that allows us to store a field element in unit space and perform an arithmetic operation in unit time. Over $\mathbb{Q}$, it is not clear if MCSP for arithmetic circuits is even decidable in the Turing machine model. But, if we confine our search to size-s circuits whose field constants are $s^{O(1)}$ bit rational numbers, then we can work with the Turing machine model.

MCSP for homogeneous depth-3 circuits. The size of a $\Sigma\Pi\Sigma$ circuit is primarily determined by its formal degree and its top fan-in, whereas the size of a homogeneous depth-3 (hom-$\Sigma\Pi\Sigma$) circuit is mainly decided by its top fan-in (the formal degree of a $\Sigma\Pi\Sigma$ circuit is the maximum

[7] i.e., the size of the output of the reduction and the output parameter s depend only on the size of the input instance. Almost all reductions that show NP-hardness of problems are natural.

[8] Sparse representations of polynomials are also used in computer algebra systems wherein the exponent vector is given in binary. As the degree is $n^{O(1)}$ in this work (except on one occasion; see the remark following Theorem 16), whether the exponent vector is given in unary or binary makes little difference.

fan-in of the middle layer of × gates). MCSP for $\Sigma\Pi\Sigma$ circuits can be defined as follows: given f and $D, s \in \mathbb{N}$, decide if there is a $\Sigma\Pi\Sigma$ circuit with formal degree bounded by D and top fan-in bounded by s that computes f. Similarly, MCSP for hom-$\Sigma\Pi\Sigma$ circuits is defined as: given a homogeneous f and $s \in \mathbb{N}$, check if there is a hom-$\Sigma\Pi\Sigma$ circuit with top fan-in at most s that computes f. In order to prove NP-hardness of $\Sigma\Pi\Sigma$-MCSP, it is *necessary* to prove NP-hardness of hom-$\Sigma\Pi\Sigma$-MCSP. The reason is: a polynomial $f(x_1, x_2, \ldots, x_n)$ has a $\Sigma\Pi\Sigma$ circuit with formal degree bounded by D and top fan-in bounded by s if and only if the homogeneous polynomial $z^D f(x_1 z^{-1}, x_2 z^{-1}, \ldots, x_n z^{-1})$ has a hom-$\Sigma\Pi\Sigma$ circuit with top fan-in bounded by s. Moreover, if the reduction in a hypothetical proof of NP-hardness of hom-$\Sigma\Pi\Sigma$-MCSP has a certain simple feature, then it would imply NP-hardness of $\Sigma\Pi\Sigma$-MCSP (see the second remark following Proposition 39). Hence, it is natural to study the hardness of hom-$\Sigma\Pi\Sigma$-MCSP first.

NP-hardness of MCSP is known for two interesting subclasses of hom-$\Sigma\Pi\Sigma$ circuits, namely depth-3 powering circuits [53] and set-multilinear $\Sigma\Pi\Sigma$ circuits [26]; the top fan-in's of circuits in these two classes correspond to Waring rank and tensor rank, respectively. Perhaps an appealing evidence in favor of NP-hardness of hom-$\Sigma\Pi\Sigma$-MCSP is a proof of NP-hardness of MCSP for a "dense" subclass of hom-$\Sigma\Pi\Sigma$ circuits. Intuitively, $\mathcal{C}$ is a *dense* subclass of hom-$\Sigma\Pi\Sigma$ circuits if every hom-$\Sigma\Pi\Sigma$ circuit can be approximated "infinitesimally closely" by circuits in $\mathcal{C}$.[9] Unfortunately, depth-3 powering circuits and set-multilinear $\Sigma\Pi\Sigma$ circuits are *not* dense inside hom-$\Sigma\Pi\Sigma$ circuits.[10] On the other hand, *orbits of homogeneous sparse polynomials* form a dense subclass of hom-$\Sigma\Pi\Sigma$ circuits.[11] It is natural to ask:

Is MCSP for orbits of homogeneous sparse polynomials NP-hard?

MCSP for orbits of homogeneous sparse polynomials is exactly the ETsparse problem on inputs that are homogeneous polynomials. The second part of Theorem 2 answers the question positively over fields of characteristic 0.

Approximating the sparse-orbit complexity. Call the smallest s_0 such that f is in the orbit of an s_0-sparse polynomial, the *sparse-orbit complexity* of f. Theorem 2 shows that sparse-orbit complexity is hard to compute in the worst case.

Is sparse-orbit complexity easy to approximate?

In Theorem 8, we show that it is NP-hard to approximate the sparse-orbit complexity of a given s-sparse polynomial (homogeneous or not) to within a $s^{\frac{1}{3}-\epsilon}$ factor for any $\epsilon \in (0, 1/3)$. For s-sparse inputs, a within s factor approximation of the sparse-orbit complexity is trivial.

[9] Formally, a subclass $\mathcal{C}$ of hom-$\Sigma\Pi\Sigma$ circuits is *dense* if there are polynomial functions $p, q : \mathbb{N} \to \mathbb{N}$ such that the following holds: For $n, d, s \in \mathbb{N}$, the coefficient vector of every n-variate degree-d polynomial computable by a size-s hom-$\Sigma\Pi\Sigma$ circuit is in the *Zariski closure* of the set of coefficient vectors of $p(nds)$-variate degree-d polynomials computable by size-$q(nds)$ circuits in $\mathcal{C}$. Here, "size" means "top fan-in".

[10] Circuits of these two classes have small read-once algebraic branching programs (ROABPs), and the class ROABP is closed under Zariski closure [18]. So, the closures of these two classes are also contained inside ROABPs. But, there are explicit $O(n)$ size hom-$\Sigma\Pi\Sigma$ circuits that require $2^{\Omega(n)}$ size ROABPs [50, 38].

[11] Every n-variate degree-d size-s hom-$\Sigma\Pi\Sigma$ circuit is a linear projection of an s-sparse degree-d homogeneous polynomial in at most sd variables. It is well known that linear projections of f are contained in the Zariski closure of the orbit of f over fields of characteristic 0 (see [50] for a proof of this fact).

1.2 ET for constant-support polynomials

ET is efficiently solvable for two special sparse polynomial families, namely the power symmetric polynomial $\mathsf{PSym} := x_1^d + \ldots + x_n^d$ [36] and the sum-product polynomial $\mathsf{SP} := \sum_{i\in[s]}\prod_{j\in[d]} x_{i,j}$ [45, 36]. What makes ET easy for these sparse polynomials? Explanations were provided in [24, 8]: SP is a read-once formula; it is also a 1-design polynomial. PSym is a 1-design polynomial, but it is also a support-1 polynomial.

Is ET easy for constant-support polynomials?

In Theorem 16, we show that checking if a given f is in the orbit of a support-6 polynomial is NP-hard; this answers the question in the negative.

1.3 Our results

We now state our results formally. The ETsparse problem is defined as follows.

▶ **Problem 1** (ETsparse)**.** *Given a polynomial $f \in \mathbb{F}[\mathbf{x}]$ in its sparse representation and an $s \in \mathbb{Z}$, check if there exists an $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$ and a $\mathbf{b} \in \mathbb{F}^{|\mathbf{x}|}$ such that $f(A\mathbf{x} + \mathbf{b})$ is s-sparse.*

Our first result, Theorem 2, shows the NP-hardness of ETsparse over any field.

▶ **Theorem 2** (ETsparse is NP-hard)**.**

1. *Let $\mathbb{F}$ be any field. There exists a deterministic polynomial-time many-one reduction from* 3-SAT *to* ETsparse *over $\mathbb{F}$.*
2. *Let $\mathrm{char}(\mathbb{F}) = 0$. There exists a deterministic polynomial-time many-one reduction from* 3-SAT *to* ETsparse *over $\mathbb{F}$ where the input polynomial in* ETsparse *is homogeneous.*

▶ Remark 3. The reduction is *natural*[12] and has the feature that a satisfying assignment can be mapped to a sparsifying invertible $A \in \{-1, 0, 1\}^{|\mathbf{x}|\times|\mathbf{x}|}$ and vice versa. So, ETsparse is NP-hard even when A is restricted to having only $\{-1, 0, 1\}$ entries.

▶ Remark 4. The authors of [15] showed the undecidability over $\mathbb{Z}$ of testing if a given f is shift equivalent to some sparse polynomial (f is shift equivalent to a polynomial g, if there exists a $\mathbf{b} \in \mathbb{F}^{|\mathbf{x}|}$ s.t $f = g(\mathbf{x} + \mathbf{b})$). However, their result does not imply the intractability of ETsparse as testing shift equivalence to a sparse polynomial is a special case of ETsparse when A is the identity map.

▶ Remark 5. Depth-3 power circuits, set-multilinear depth-3 circuits, and shifted sparse polynomials are all contained inside ROABPs. So, these models admit polynomial-time (improper) learning algorithms [9, 40] and quasi-polynomial-time hitting sets [1, 19]. Orbits of sparse polynomials require exponential size ROABPs [50]; we cannot expect to improperly learn them via ROABPs. Theorem 2 suggests that proper learning orbits of sparse polynomials is likely hard. Nonetheless, there is a quasi-polynomial time hitting set for orbits of sparse polynomials [45, 50].

▶ Remark 6. We believe that with some more effort, the second part of Theorem 2 can be proven over fields of finite characteristics as well. See the last remark in Section 3.4.

We prove Theorem 2 in Section 3. Next, we define the gap version of ETsparse.

▶ **Problem 7** (α-gap-ETsparse)**.** *Let $\alpha > 1$ be a parameter. Given a polynomial $f \in \mathbb{F}[\mathbf{x}]$ in its sparse representation and an integer s_0, output:*

- *YES, if there exist an $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$ and $\mathbf{b} \in \mathbb{F}$ such that $f(A\mathbf{x} + \mathbf{b})$ is s_0-sparse.*
- *NO, if for all $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$ and $\mathbf{b} \in \mathbb{F}$, $f(A\mathbf{x} + \mathbf{b})$ has sparsity at least αs_0.*

[12] unless $\mathrm{char}(\mathbb{F}) = 2$. See the full version for details.

Our second result, Theorem 8, shows that α-gap-ETsparse is NP-hard for $\alpha = s^{\frac{1}{3}-\epsilon}$, where s is the sparsity of the input polynomial f and $\epsilon \in (0, \frac{1}{3})$ is an arbitrary constant. Theorem 8 is proven in Section 4. From Theorem 8, we get Corollary 11 which states that $s^{\frac{1}{3}-\epsilon}$ factor approximation of the sparse-orbit complexity of an s-sparse polynomial is NP-hard.

▶ **Theorem 8** ($s^{\frac{1}{3}}$-gap-ETsparse is NP-hard)**.** *Let $\epsilon \in (0, \frac{1}{3})$ be an arbitrary constant.*

1. *Let $\mathbb{F}$ be any field. There exists a deterministic polynomial-time many-one reduction from* 3-SAT *to* $s^{\frac{1}{3}-\epsilon}$-gap-ETsparse *over $\mathbb{F}$ where the input polynomial in* $s^{\frac{1}{3}-\epsilon}$-gap-ETsparse *is s-sparse.*
2. *Let* $\mathrm{char}(\mathbb{F}) = 0$*. There exists a deterministic polynomial-time many-one reduction from* 3-SAT *to* $s^{\frac{1}{3}-\epsilon}$-gap-ETsparse *over $\mathbb{F}$ where the input polynomial in* $s^{\frac{1}{3}-\epsilon}$-gap-ETsparse *is homogeneous and s-sparse.*

▶ Remark 9. With a more careful analysis, the constant $\frac{1}{3}$ in $s^{\frac{1}{3}-\epsilon}$ may be improved.

▶ Remark 10. Interestingly, the above results are obtained without invoking the celebrated PCP theorem [7, 6, 17].

▶ **Corollary 11.** *Let $0 < \epsilon < \frac{1}{3}$ be an arbitrary constant.*

1. *Let $\mathbb{F}$ be any field. It is* NP*-hard to compute $s^{\frac{1}{3}-\epsilon}$ factor approximation of the sparse-orbit complexity when the input is an s-sparse polynomial over $\mathbb{F}$.*
2. *Let* $\mathrm{char}(\mathbb{F}) = 0$*. It is* NP*-hard to compute $s^{\frac{1}{3}-\epsilon}$ factor approximation of the sparse-orbit complexity when the input is an s-sparse homogeneous polynomial over $\mathbb{F}$.*

Now, we formally define the support of a polynomial.

▶ **Definition 12** (Support of a polynomial)**.** *For a monomial $\mathbf{x}^{\boldsymbol{\alpha}}$, where $\boldsymbol{\alpha}$ is the exponent vector, the support of $\mathbf{x}^{\boldsymbol{\alpha}}$,* $\mathrm{Supp}(\mathbf{x}^{\boldsymbol{\alpha}})$*, is the number of variables with non-zero exponent. The support of a polynomial f,* $\mathrm{Supp}(f)$*, is the maximum support size over all the monomials of f.*

Thus, a polynomial has support σ if there exists a monomial with support σ and no other monomial has support $> \sigma$. The ET problem for constant-support polynomials and a stronger version of it are defined next (henceforth, σ is assumed to be a constant).

▶ **Problem 13** (ETsupport)**.** *Given a polynomial $f \in \mathbb{F}[\mathbf{x}]$ in its sparse representation and an integer σ, check if there exists an $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$ such that $\mathrm{Supp}(f(A\mathbf{x})) \leq \sigma$.*

▶ **Problem 14** (($\sigma + 1$)-to-σ ETsupport)**.** *Given a polynomial $f \in \mathbb{F}[\mathbf{x}]$ with support $\sigma + 1$ in its sparse representation, check if there exists an $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$ such that $\mathrm{Supp}(f(A\mathbf{x})) \leq \sigma$.*

▶ Remark 15. Unlike ETsparse, checking if f is in the *orbit* of a constant-support polynomial is the same as checking if f is equivalent to a constant-support polynomial. This follows from the observation that $\mathrm{Supp}(f(\mathbf{x})) = \mathrm{Supp}(f(\mathbf{x} + \mathbf{b}))$ for any $\mathbf{b} \in \mathbb{F}^{|\mathbf{x}|}$.

Our third and last result, Theorem 16, shows that ETsupport and ($\sigma + 1$)-to-σ ETsupport are NP-hard. We prove Theorem 16 in Section 5.

▶ **Theorem 16** (ETsupport is NP-hard)**.** *Let $\sigma \geq 6$ be a constant and $\mathbb{F}$ be a field with* $\mathrm{char}(\mathbb{F}) = 0$ *or $> \sigma + 1$. There is a deterministic polynomial-time many-one reduction from* 3-SAT *to* ETsupport *over $\mathbb{F}$. In particular,* 3-SAT *reduces to* ($\sigma + 1$)-to-σ ETsupport *in deterministic polynomial time.*

▶ Remark 17. Over fields of finite characteristic, it is assumed that the exponent vectors corresponding to the monomials of the input polynomial are given in binary.

We prove Theorems 2, 8 and 16 by direct reductions from 3-SAT, and at the beginning of Sections 3, 4 and 5, we give proof sketches of the respective reductions.

2 Preliminaries

2.1 Definitions and notations

For $n, a, b \in \mathbb{N}$, $[n]$ denotes the set $\{1, 2 \ldots, n\}$ and $[a, b]$ denotes the integers from a to b, both inclusive. A polynomial is *homogeneous* if all its monomials have the same total degree. The set of invertible linear transforms in n variables over a field $\mathbb{F}$ is denoted by $\mathrm{GL}(n, \mathbb{F})$. For a polynomial $f \in \mathbb{F}[\mathbf{x}]$, the action of a linear transform $A \in \mathbb{F}^{|\mathbf{x}|\times|\mathbf{x}|}$ on its variables is denoted by $f(A\mathbf{x})$ as well as by $A(f)$. The sparsity of a polynomial f, denoted as $\mathcal{S}(f)$, is the number of monomials in f with non-zero coefficients. For a polynomial f, $\mathrm{var}(f)$ denotes the set of variables that occur in at least one monomial of f. We have used the notation $f \sim g$ earlier to denote $f = g(A\mathbf{x} + \mathbf{b})$. Henceforth, we will ignore the translation vector $\mathbf{b}$ in the main body of the discussion for simplicity but mention the necessary changes in the proofs or point to appropriate sections when translations are involved. Thus, for polynomials f and g, $f \sim g$ will mean $f(\mathbf{x}) = g(A\mathbf{x})$ where $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$.[13] Similarly, the orbit of a polynomial f will now denote the set $\{f(A\mathbf{x}), A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})\}$.

▶ **Definition 18** (Degree separated polynomials). *Polynomials f and g are* ***degree separated*** *if no monomial of f has the same degree as a monomial of g. Similarly, f and g are* ***degree separated with respect to a variable*** *x if no monomial of f has the same x-degree as a monomial of g.*[14]

2.2 Algebraic preliminaries

The proofs of the observations and claims in this section can be found in the full version.

▶ **Observation 19.** *Let f and g be polynomials such that $f \sim g$. Then, f and g have the same set of degrees*[15] *for the monomials. Thus, if f and g are degree separated, then $f \not\sim g$.*

▶ **Observation 20.** *If f and g are degree separated (or degree separated with respect to some variable), then $\mathcal{S}(f + g) = \mathcal{S}(f) + \mathcal{S}(g)$.*

▶ **Observation 21.** *If f and g are degree separated, $f_1 \sim f$ and $g_1 \sim g$, then $\mathcal{S}(f_1 + g_1) = \mathcal{S}(f_1) + \mathcal{S}(g_1)$.*

Observation 22 analyzes the sparsity of powers of linear forms. Observation 23 is a special case of Observation 22 and is stated separately because it is simpler and is invoked many times. Observation 24 analyzes the sparsity of powers of affine forms.

▶ **Observation 22.** *Let ℓ be a m-variate linear form*[16] *and $d \in \mathbb{N}$. If $\mathrm{char}(\mathbb{F}) = 0$, $\mathcal{S}(\ell^d) = \binom{d+m-1}{m-1}$ and if $\mathrm{char}(\mathbb{F}) = p$, $\mathcal{S}(\ell^d) = \prod_{i=0}^{k} \binom{e_i+m-1}{m-1}$, where $d = \sum_{i=0}^{k} e_i p^i$, $e_i \in [0, p-1]$.*

▶ **Observation 23.** *If $\mathrm{char}(\mathbb{F}) = 0$ and ℓ is a linear form in exactly two variables, then $\mathcal{S}(\ell^d) = d+1$. The result holds for $\mathrm{char}(\mathbb{F}) = p$ fields if $p > d$ or if $d = p^k - 1$ for some $k \in \mathbb{N}$. Further, if ℓ is a linear form in at least two variables and d is as before, then $\mathcal{S}(\ell^d) \geq d + 1$.*

[13] Note $\sim$ is an equivalence relation under this definition.

[14] The degree of a monomial means its total degree and the degree of a polynomial f is the maximum degree amongst all monomials in f. The x-degree of a monomial is the degree of the variable x in the monomial.

[15] The set of degrees is the set of distinct degrees of all the monomials in the polynomial. For example, the set of degrees of $f(x_1, x_2) = x_1^2 + x_1x_2 + 4x_2$ is $\{2, 1\}$.

[16] A linear form is a homogeneous degree one polynomial. An affine form is a degree one polynomial.

▶ **Observation 24.** *Let $h = \ell + c_0$, where ℓ is a linear form in at least one variable and $c_0 \in \mathbb{F}\backslash\{0\}$, then $\mathcal{S}(h^d) \geq \mathcal{S}(\ell^d) + 1$. More precisely, $\mathcal{S}(h^d) \geq d + 1$ holds if* $\mathrm{char}(\mathbb{F}) = 0$ *or if* $\mathrm{char}(\mathbb{F}) = p$ *and $p > d$ or $d = p^k - 1$ for some $k \in \mathbb{N}$.*

Claim 25 analyzes the sparsity of polynomials divisible by a power of some linear form in at least two variables and is used to prove part two of Theorems 2 and 8. Claim 26 analyzes the support of monomials under invertible linear transforms and is used to prove Theorem 16.

▷ Claim 25. Let $\mathrm{char}(\mathbb{F}) = 0$. If $f \in \mathbb{F}[\mathbf{x}]$ is a non-zero polynomial divisible by ℓ^d for some linear form ℓ in at least two variables, then $\mathcal{S}(f) \geq d + 1$.

▷ Claim 26. Let $\sigma, d, n \in \mathbb{N}$, $d \geq \sigma$, $f = (x_1 \cdots x_n)^d$, and $\ell_1, \ldots, \ell_n$ be linearly independent linear forms in $x_1, \ldots, x_n$. If $|\cup_{i=1}^{n} \mathrm{var}(\ell_i)| \geq \sigma$ and $g := f(\ell_1 \cdots \ell_n)$, then $\mathrm{Supp}(g) \geq \sigma$. The claim holds if $\mathrm{char}(\mathbb{F}) = 0$, or $\mathrm{char}(\mathbb{F}) = p$ with $p > d$, or $p > \sigma$ and $d = p^k - 1$ for some $k \in \mathbb{N}$.

3 NP-hardness of ETsparse

In this section, we prove Theorem 2 over $\mathrm{char}(\mathbb{F}) = 0$ fields without translations for easy understanding.[17] The full version contains the proofs of the lemmas and the observations in this section, and the reduction over $\mathrm{char}(\mathbb{F}) > 0$ fields and also when translations are allowed.

Proof sketch. The reduction maps each variable and clause of a 3-CNF[18] ψ to distinct degree separated polynomials which, summed together, give the polynomial f. As the summands are degree separated, the sparsity of f under invertible transforms can be analyzed by doing so for individual polynomials. The degrees are chosen such that f is equivalent to an s-sparse polynomial (for a suitable sparsity parameter s) if and only if $\psi \in$ 3-SAT.

3.1 Constructing f and s

Let ψ be a 3-CNF in variables $\mathbf{x} := \{x_1, x_2 \ldots x_n\}$ and m clauses:

$$\psi = \wedge_{k=1}^{m} \vee_{j \in C_k} (x_j \oplus a_{k,j}),$$

where C_k denotes the set of indices of the variables in the k^{th} clause and $a_{k,j} \in \{0, 1\}$. Let $\mathbf{y} := \{y_1, y_2 \ldots y_n\}$, x_0 be a new variable and $\mathbf{z} := \{x_0\} \sqcup \mathbf{x} \sqcup \mathbf{y}$. For $d_1, d_2, d_3, d_4 \in \mathbb{N}$, consider the following polynomials:

- Corresponding to variable x_i, where $i \in [n]$, define $Q_i(\mathbf{z})$ as:

$$Q_i(\mathbf{z}) := Q_{i,1}(\mathbf{z}) + Q_{i,2}(\mathbf{z}) + Q_{i,3}(\mathbf{z}), \text{ where}$$
$$Q_{i,1}(\mathbf{z}) := x_0^{(3i-2)d_1} x_i^{d_2}, \; Q_{i,2}(\mathbf{z}) := x_0^{(3i-1)d_1}(y_i + x_i)^{d_3} \text{ and } Q_{i,3}(\mathbf{z}) := x_0^{3id_1}(y_i - x_i)^{d_3}.$$

- For the k^{th} clause, $k \in [m]$, define $R_k(\mathbf{z}) := x_0^{(3n+k)d_1} \prod_{j \in C_k} (y_j + (-1)^{a_{k,j}} x_j)^{d_4}$.

[17] Note that for f and g two homogeneous polynomials, $f(\mathbf{x}) = g(A\mathbf{x} + \mathbf{b})$ implies $f(\mathbf{x}) = g(A\mathbf{x})$, where $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$ and $\mathbf{b} \in \mathbb{F}^{|\mathbf{x}|}$. Hence, it suffices to prove part 2 of Theorem 2 without translations.

[18] We assume, without loss of generality, that each clause of a 3-CNF has 3 distinct variables. This can be achieved by introducing extra variables for clauses with < 3 variables.

Define $s := 1 + n(3 + d_3) + m(d_4 + 1)^2$ and the polynomial f as:

$$f(\mathbf{z}) := x_0^{d_1} + \sum_{i=1}^{n} Q_i(\mathbf{z}) + \sum_{k=1}^{m} R_k(\mathbf{z}). \tag{1}$$

The following conditions are imposed on the d_i's:

$$d_1 \geq \max(s, d_2 + 1),\ d_2 \geq 2d_3,\ d_3 \geq m(d_4 + 1)^2 + 1, \text{ and } d_4 \geq m. \tag{2}$$

Set $d_4 := m$, $d_3 := m(m + 1)^2 + 1 = O(m^3)$, $d_2 = 2m(m + 1)^2 + 2 = O(m^3)$ and $d_1 = 1 + n(4 + m(m + 1)^2) + m(m + 1)^2 = O(nm^3)$. Then $s = O(nm^3)$ and the d_i's satisfy the conditions of (2), under which the following observations hold.

▶ **Observation 27.** *For $i \in [n]$, $k \in [m]$, the polynomials $x_0^{d_1}$, $Q_{i,1}(\mathbf{z})$, $Q_{i,2}(\mathbf{z})$, $Q_{i,3}(\mathbf{z})$ and $R_k(\mathbf{z})$ are degree separated from one another. Also, $Q_i(\mathbf{z})$ is degree separated from $Q_j(\mathbf{z})$, for $i, j \in [n]$ and $i \neq j$. Similarly, $R_k(\mathbf{z})$ is degree separated from $R_l(\mathbf{z})$ for $k, l \in [m]$ and $k \neq l$.*

▶ **Observation 28.** *The degree of f is $(3n + m)d_1 + 3d_4 = (mn)^{O(1)}$, $\mathcal{S}(f(\mathbf{z})) = 1 + n(2d_3 + 3) + m(d_4 + 1)^3$ and $\mathrm{Supp}(f) = 7$.*

3.2 The forward direction

Proposition 29 shows how a satisfiable ψ implies the existence of an invertible A, such that $\mathcal{S}(f(A\mathbf{z})) \leq s$ by constructing A from a satisfying assignment $\mathbf{u} \in \{0, 1\}^n$ of ψ.

▶ **Proposition 29.** *Let $\mathbf{u} \in \{0, 1\}^n$ be such that $\psi(\mathbf{u}) = 1$. Then $\mathcal{S}(f(A\mathbf{z})) \leq s$ for A as:*

$$A : x_0 \mapsto x_0, x_i \mapsto x_i, y_i \mapsto y_i + (-1)^{u_i} x_i, \quad \forall i \in [n]. \tag{3}$$

Proof. It follows from the definition of f in (1), Observations 27 and 21 that

$$\mathcal{S}(f(A\mathbf{z})) = \mathcal{S}(A(x_0^{d_1})) + \sum_{i=1}^{n} \mathcal{S}(Q_i(A\mathbf{z})) + \sum_{k=1}^{m} \mathcal{S}(R_k(A\mathbf{z})).$$

Thus, it suffices to analyze the sparsity of $A(x_0^{d_1}), Q_i(A\mathbf{z})$'s and $R_k(A\mathbf{z})$'s. Now, $\mathcal{S}(A(x_0^{d_1})) = 1$ as $A(x_0^{d_1}) = x_0^{d_1}$. We now analyze $\mathcal{S}(Q_i(A\mathbf{z}))$ for $i \in [n]$. If $u_i = 0$, then

$$Q_{i,1}(A\mathbf{z}) = x_0^{(3i-2)d_1} x_i^{d_2},\ Q_{i,2}(A\mathbf{z}) = x_0^{(3i-1)d_1}(y_i + 2x_i)^{d_3} \text{ and } Q_{i,3}(A\mathbf{z}) = x_0^{3id_1} y_i^{d_3}.$$

If $u_i = 1$, then

$$Q_{i,1}(A\mathbf{z}) = x_0^{(3i-2)d_1} x_i^{d_2},\ Q_{i,2}(A\mathbf{z}) = x_0^{(3i-1)d_1} y_i^{d_3} \text{ and } Q_{i,3}(A\mathbf{z}) = x_0^{3id_1}(y_i - 2x_i)^{d_3}.$$

By Observation 23 (for linear forms in two variables over $\mathrm{char}(\mathbb{F}) = 0$ fields), if $u_i = 0$ then $\mathcal{S}(Q_{i,2}(A\mathbf{z})) = d_3 + 1$ and $\mathcal{S}(Q_{i,3}(A\mathbf{z})) = 1$ and, if $u_i = 1$ then $\mathcal{S}(Q_{i,2}(A\mathbf{z})) = 1$ and $\mathcal{S}(Q_{i,3}(A\mathbf{z})) = d_3 + 1$. In either case, by Observations 27 and 21,

$$\mathcal{S}(Q_i(A\mathbf{z})) = \mathcal{S}(Q_{i,1}(A\mathbf{z})) + \mathcal{S}(Q_{i,2}(A\mathbf{z})) + \mathcal{S}(Q_{i,3}(A\mathbf{z})) = d_3 + 3.$$

For the k^{th} clause, $k \in [m]$, the action of A on the corresponding polynomial R_k is:

$$R_k(A\mathbf{z}) = x_0^{(3n+k)d_1} \prod_{j \in C_k} (y_j + ((-1)^{a_{k,j}} + (-1)^{u_j})x_j)^{d_4}.$$

As the multiplicands in $R_k(A\mathbf{z})$ do not share any variables, $\mathcal{S}(R_k(A\mathbf{z}))$ is the product of the sparsity of the multiplicands. Since $\psi(\mathbf{u}) = 1$, therefore in the k^{th} clause there exists $j \in C_k$ such that $a_{k,j} \neq u_j$. For that j, $(y_j + ((-1)^{a_{k,j}} + (-1)^{u_j})x_j)^{d_4} = y_j^{d_4}$. As at least one literal is true in every clause under $\mathbf{u}$, $\mathcal{S}(R_k(A\mathbf{z})) \leq (d_4+1)^2$ using Observation 23. Thus,

$$\begin{aligned}\mathcal{S}(f(A\mathbf{z})) &= \mathcal{S}(A(x_0^{d_1})) + \sum_{i=1}^{n} \mathcal{S}(Q_i(A\mathbf{z})) + \sum_{k=1}^{m} \mathcal{S}(R_k(A\mathbf{z})) \\ &\leq 1 + n(d_3+3) + m(d_4+1)^2 = s. \end{aligned}$$ ◀

3.3 The reverse direction

Now, we show that $(f, s) \in$ ETsparse implies $\psi \in$ 3-SAT by showing that the permuted and scaled versions of the transform of (3) form all the viable sparsifying invertible linear transforms. This is where the constraints on the d_i's are used. So, let $A \in \mathrm{GL}(|\mathbf{z}|, \mathbb{F})$ be such that $\mathcal{S}(f(A\mathbf{z})) \leq s$. Lemma 30 shows that $A(x_0)$ is just a variable by leveraging $d_1 \geq s$.

▶ **Lemma 30.** *Without loss of generality, $A(x_0) = x_0$.*

The proof of Lemma 31 uses $d_2 \geq 2d_3$ while that of Lemma 32 uses $d_3 \geq m(d_4+1)^2 + 1$.

▶ **Lemma 31.** *For any invertible A and $i \in [n]$:*

$$\mathcal{S}(Q_i(A\mathbf{z})) = \mathcal{S}(Q_{i,1}(A\mathbf{z})) + \mathcal{S}(Q_{i,2}(A\mathbf{z})) + \mathcal{S}(Q_{i,3}(A\mathbf{z})) \geq d_3 + 3,$$

where Q_i, $Q_{i,1}$, $Q_{i,2}$ and $Q_{i,3}$ are as defined in Section 3.1. Equality holds if and only if $A(x_i) = X_i$ and $A(y_i) = Y_i + (-1)^{u_i} X_i$ for some scaled variables $X_i, Y_i \in \mathbf{z}$ and $u_i \in \{0,1\}$. Further, if $\mathcal{S}(Q_i(A\mathbf{z})) \neq d_3 + 3$, then $\mathcal{S}(Q_i(A\mathbf{z})) \geq 2d_3 + 3$.

▶ **Lemma 32.** *Under the given A, $\mathcal{S}(Q_i(A\mathbf{z})) = d_3 + 3$ holds for all $i \in [n]$.*

Lemmas 30, 31 and 32 together show that A is a permuted scaled version of the transform of (3). We can assume A to be as described in (3) without loss of generality as permutation and non-zero scaling of variables do not affect the sparsity of a polynomial. Proposition 33 shows how a satisfying assignment can be formed from A using $d_4 \geq m$.

▶ **Proposition 33.** *For A as given in (3), $\mathbf{u} = (u_1, \ldots, u_n)$ is a satisfying assignment for ψ.*

Proof. Suppose not; then there exists $k \in [m]$ such that the k^{th} clause, $\vee_{j \in C_k}(x_j \oplus a_{k,j})$, in ψ is unsatisfied. Since this clause is unsatisfied, $u_j = a_{k,j}$ for all $j \in C_k$. Thus, $R_k(A\mathbf{z}) = x_0^{(3n+k)d_1} \prod_{j \in C_k}(y_j \pm 2x_j)^{d_4}$, where R_k is as defined in Section 3.1, and $\mathcal{S}(R_k(A\mathbf{z})) = (d_4+1)^3 \geq (m+1)(d_4+1)^2$ by Observation 23, the fact that $R_k(A\mathbf{z})$ is a product of linear forms not sharing variables, and the condition $d_4 \geq m$. By the definition of f and s in Section 3.1, Observations 27 and 21, it holds that

$$\begin{aligned}\mathcal{S}(f(A\mathbf{z})) &\geq \mathcal{S}(A(x_0)^{d_1}) + \sum_{i=1}^{n} \mathcal{S}(Q_i(A\mathbf{z})) + \mathcal{S}(R_k(A\mathbf{z})) \\ &\geq 1 + n(3+d_3) + m(d_4+1)^2 + (d_4+1)^2 = s + (d_4+1)^2 > s, \end{aligned}$$

a contradiction. Thus, $\mathbf{u}$ is a satisfying assignment for ψ. ◀

3.4 Homogeneous case: Proof of Part 2 of Theorem 2

We show a modification of the construction in Section 3.1 which, along with arguments similar to those in Sections 3.2 and 3.3, can be used to prove Theorem 2 for homogeneous polynomials over $\mathrm{char}(\mathbb{F}) = 0$ fields. Because the polynomials are homogeneous, we cannot use degree separation like in the non-homogeneous case. Instead, we introduce a new variable y_0 and redefine $Q_i(\mathbf{z})$ and $R_k(\mathbf{z})$ of Section 3.1 so that:

1. Each polynomial is homogeneous with the same degree and is divisible by $x_0^{d_1}$ and $y_0^{d_1}$.
2. Each polynomial has distinct y_0-degree and if P_1 and P_2 are polynomials where P_1 has a higher y_0-degree than P_2, then the y_0-degree of P_1 is greater than the degree of any variable (except possibly x_0) in P_2.

The divisibility condition ensures that both x_0 and y_0 map to scaled variables (see Lemma 37 and its proof), and the second condition induces a degree separation of the polynomials with respect to y_0 (see Observation 34 and Lemma 38). Formally, let x_0, $\mathbf{x}$ and $\mathbf{y}$ be as defined in Section 3.1 and y_0 be a new variable. Define $\mathbf{z} := \mathbf{x} \sqcup \mathbf{y} \sqcup \{x_0\} \sqcup \{y_0\}$. Let $d_1, d_2, d_3, d_4 \in \mathbb{N}$. Consider the following polynomials:

1. For each variable x_i, $i \in [n]$, let $Q_i(\mathbf{z}) := Q_{i,1}(\mathbf{z}) + Q_{i,2}(\mathbf{z}) + Q_{i,3}(\mathbf{z})$, where

$$Q_{i,1}(\mathbf{z}) := x_0^{d_1(3n-3i+6)-d_2} y_0^{d_1(3i+m-2)} x_i^{d_2}, \; Q_{i,2}(\mathbf{z}) := x_0^{d_1(3n-3i+5)-d_3} y_0^{d_1(3i+m-1)} (y_i + x_i)^{d_3}$$
$$Q_{i,3}(\mathbf{z}) := x_0^{d_1(3n-3i+4)-d_3} y_0^{d_1(3i+m)} (y_i - x_i)^{d_3}.$$

2. For the k^{th} clause, $k \in [m]$, let $R_k(\mathbf{z}) := x_0^{d_1(3n+m+4-k)-3d_4} y_0^{d_1 k} \prod_{j\in C_k} (y_j + (-1)^{a_{k,j}} x_j)^{d_4}$.

Define $s := 1 + n(d_3 + 3) + m(d_4 + 1)^2$ as before and impose the conditions of (2) on the d_i's. Using the conditions on the d_i's, it is easy to verify that the individual degrees of x_0 and y_0 in every polynomial defined above is at least d_1. Define f as:

$$f(\mathbf{z}) := x_0^{3d_1} y_0^{d_1(3n+m+1)} + \sum_{i=1}^{n} Q_i(\mathbf{z}) + \sum_{k=1}^{m} R_k(\mathbf{z}). \tag{4}$$

Clearly, f is a homogeneous polynomial of degree $(3n + m + 4)d_1$ and is divisible by $x_0^{d_1}$ and $y_0^{d_1}$. Further, we have the following observations under the constraints of (2).

▶ **Observation 34.** *For all $i \in [n]$, $k \in [m]$, the polynomials $x_0^{3d_1} y_0^{d_1(3n+m+1)}$, $Q_{i,1}(\mathbf{z})$, $Q_{i,2}(\mathbf{z})$, $Q_{i,3}(\mathbf{z})$ and $R_k(\mathbf{z})$ are degree separated with respect to y_0 from one another. Also, $Q_i(\mathbf{z})$ is degree separated with respect to y_0 from other $Q_j(\mathbf{z})$'s, for $i, j \in [n]$ and $i \neq j$. Similarly, $R_k(\mathbf{z})$ is degree separated with respect to y_0 from $R_l(\mathbf{z})$ for $k, l \in [m]$ and $k \neq l$.*

▶ **Observation 35.** *$\mathcal{S}(f(\mathbf{z})) = 1 + n(2d_3 + 3) + m(d_4 + 1)^3$ and $\mathrm{Supp}(f) = 8$.*

The forward direction. Let $\mathbf{u} \in \{0, 1\}^n$ be such that $\psi(\mathbf{u}) = 1$ and f, as described in (4), be the polynomial corresponding to ψ. Proposition 36 shows how to construct a sparsifying transform from $\mathbf{u}$. The proof of Proposition 36 is very similar to that of Proposition 29.

▶ **Proposition 36.** *$\mathcal{S}(f(A\mathbf{z})) \leq s$ where $A \in \mathrm{GL}(|\mathbf{z}|, \mathbb{F})$ is as follows:*

$$A : y_0 \mapsto y_0, \; x_0 \mapsto x_0, \; x_i \mapsto x_i, \; y_i \mapsto y_i + (-1)^{u_i} x_i \quad i \in [n]. \tag{5}$$

The reverse direction. Let $\mathcal{S}(f(A\mathbf{z})) \leq s$ for some $A \in \mathrm{GL}(|\mathbf{z}|, \mathbb{F})$. Lemma 37, the proof of which requires Claim 25, shows that $A(x_0)$ and $A(y_0)$ have only one variable. With this, Lemma 38 shows that the summands of $f(A\mathbf{z})$ must be degree separated with respect to y_0.

▶ **Lemma 37.** *Without loss of generality, $A(x_0) = x_0$ and $A(y_0) = y_0$.*

▶ **Lemma 38.** *For all $i \in [n]$, $k \in [m]$, the polynomials $x_0^{3d_1} y_0^{d_1(3n+m+1)}$, $Q_{i,1}(A\mathbf{z})$, $Q_{i,2}(A\mathbf{z})$, $Q_{i,3}(A\mathbf{z})$ and $R_k(A\mathbf{z})$ are degree separated from one another with respect to y_0. Also, $Q_i(A\mathbf{z})$ is degree separated with respect to y_0 from other $Q_j(A\mathbf{z})$'s, for $i, j \in [n]$ and $i \neq j$. Similarly, $R_k(A\mathbf{z})$ is degree separated with respect to y_0 from $R_l(A\mathbf{z})$ for $k, l \in [m]$ and $k \neq l$.*

$$\therefore \quad \mathcal{S}(f(A\mathbf{z})) = \mathcal{S}(x_0^{3d_1} y_0^{d_1(3n+m+1)}) + \sum_{i=1}^{n} \mathcal{S}(Q_i(A\mathbf{z})) + \sum_{k=1}^{m} \mathcal{S}(R_k(A\mathbf{z})), \text{ by Lemma 38.}$$

Lemmas 31 and 32 then hold with slight modification to their proofs, which, along with Lemma 37, show that A is a permuted scaled version of the transform of (5). Proposition 39 then holds and can be proven similarly to Proposition 33.

▶ **Proposition 39.** *For A as given in (5), $\mathbf{u} = (u_1, \ldots, u_n)$ is a satisfying assignment for ψ.*

▶ Remark 40. In the definition of f in Section 3.1 and this section, an extra summand is present besides Q_i's and R_k's. If $\text{char}(\mathbb{F}) = 0$, we can drop the summand by using Claim 25 and suitably modifying f, the current parameters and arguments to make the reduction work. We preserve the extra summand here for two reasons: One, for ease of understanding, because the definition of f in (1) is similar to that in (4). Two, in the non-homogeneous case, the extra summand proves useful in showing the reduction over finite characteristic fields, where Claim 25 does not hold, and thus it may also prove useful in showing the reduction for homogeneous polynomials over such fields.

▶ Remark 41. A feature of the reduction is that we can assume that the output polynomial is of the form $w^D f(\mathbf{z})$, where $w \notin \mathbf{z}$. This can be achieved by multiplying the output polynomial f of the current reduction by w^D, where D is greater than the sparsity parameter s in the reduction. If a proof of NP-hardness of hom-$\Sigma\Pi\Sigma$-MCSP has this feature, then it would imply NP-hardness of $\Sigma\Pi\Sigma$-MCSP (via a homogenization trick).

▶ Remark 42. We believe Claim 25 (used to prove Lemma 37) can be modified for finite characteristic fields, using which the argument in this section can be extended to such fields.

4 NP-hardness of α-gap-ETsparse

In this section, we prove parts 1 and 2 of Theorem 8 over $\text{char}(\mathbb{F}) = 0$ fields when no translations are involved. The full version contains the proofs of the lemmas in this section and that of the finite characteristic case with translations allowed for part 1 of Theorem 8.

Proof sketch. For a 3-CNF ψ, we carefully analyze the sparsity of the corresponding polynomial f, as defined in (1) with the constraints of (2) and (6), under all $A \in \text{GL}(|\mathbf{z}|, \mathbb{F})$. For $\psi \in \overline{\text{3-SAT}}$, Lemma 43 shows lower bounds on $\mathcal{S}(f(A\mathbf{z}))$ for any $A \in \text{GL}(|\mathbf{z}|, \mathbb{F})$. For $\psi \in \text{3-SAT}$, by Proposition 29, there exists $A \in \text{GL}(|\mathbf{z}|, \mathbb{F})$ such that $\mathcal{S}(f(A\mathbf{z})) \leq s_0 := 1 + n(d_3 + 3) + m(d_4 + 1)^2$. Proposition 44 compares the sparsities for satisfiable and unsatisfiable ψ's and shows α-gap-ETsparse is NP-hard using Lemma 43 and the conditions in (6).

$$d_1 = d_2 + 1, \; d_2 = d_3^2 + 1, \; d_3 = m(d_4 + 1)^2 + 1, \; d_4 \geq 4mn. \tag{6}$$

▶ **Lemma 43.** *Let $\psi \in \overline{\text{3-SAT}}$, f be as defined in* (1) *corresponding to ψ and $A \in \mathrm{GL}(|\mathbf{z}|, \mathbb{F})$.*

1. *If $A(x_0)$ is a linear form in at least 2 variables, $\mathcal{S}(f(A\mathbf{z})) \geq d_1 + 1$.*
2. *If A is not as in item 1 and $A(x_j)$ is a linear form in at least 2 variables for some $j \in [n]$, then $\mathcal{S}(f(A\mathbf{z})) \geq d_2 + 1$.*
3. *If A is not as in items 1 and 2 and $\mathcal{S}(A(y_j + x_j)) \geq 3$ or $\mathcal{S}(A(y_j - x_j)) \geq 3$ for some $j \in [n]$, then $\mathcal{S}(f(A\mathbf{z})) \geq \frac{d_3^2+3d_3+2}{2}$.*
4. *If A is not of the form described in the previous three cases, then $\mathcal{S}(f(A\mathbf{z})) \geq (d_4 + 1)^3$.*

▶ **Proposition 44.** *α-gap-ETsparse is NP-hard for s-sparse polynomial inputs over $\mathbb{F}$ and $\alpha = s^{1/3-\epsilon}$, where $\epsilon \in (0, 1/3)$ is an arbitrary constant.*

Proof. If $\psi \in$ 3-SAT, then $\mathcal{S}(f(A\mathbf{z})) \leq s_0$ where A is as described in (3). If $\psi \in \overline{\text{3-SAT}}$, then it follows from Lemma 43 that for any $A \in \mathrm{GL}(|\mathbf{z}|, \mathbb{F})$:

$$\mathcal{S}(f(A\mathbf{z})) \geq \min\left(d_1 + 1, d_2 + 1, \frac{d_3^2 + 3d_3 + 2}{2}, (d_4 + 1)^3\right).$$

The constraints imposed in (6) ensure that $(d_4+1)^3$ is the minimum. As $d_3 = m(d_4+1)^2+1$, therefore $s_0 = 1 + n(d_3 + 3) + m(d_4 + 1)^2 \leq 3nd_3 = 3mn(d_4 + 1)^2 + 3n \leq 4mn(d_4 + 1)^2$. Thus, the gap in the sparsities of the YES instances and the NO instances is

$$\frac{(d_4 + 1)^3}{s_0} \geq \frac{(d_4 + 1)^3}{4mn(d_4 + 1)^2} = \frac{d_4 + 1}{4mn}.$$

Also, as $d_4 \geq 4mn$, $\mathcal{S}(f) = s \leq 2m(d_4 + 1)^3 \implies d_4 + 1 \geq (\frac{s}{2m})^{1/3}$. Then, the gap is

$$\frac{(d_4 + 1)^3}{s_0} \geq \frac{d_4 + 1}{4mn} \geq \frac{s^{1/3}}{2^{1/3}4m^{4/3}n}.$$

Let $\epsilon \in (0, 1/3)$ be an arbitrary constant. The parameter d_4, which determines s, can be chosen a sufficiently large polynomial function in m and n such that $2^{1/3}4m^{4/3}n \leq s^{\epsilon}$. Hence, the gap is at least $s^{1/3-\epsilon}$. Thus, 3-SAT reduces to α-gap-ETsparse for $\alpha = s^{1/3-\epsilon}$. ◀

Homogeneous polynomials over $\mathrm{char}(\mathbb{F}) = 0$ fields. We now consider the polynomial f as defined in (4) for ψ with the constraints of (6). For $\psi \in$ 3-SAT, $\mathcal{S}(f(A\mathbf{z})) \leq s_0$ where A is as described in (5) and s_0 is as defined earlier. For $\psi \in \overline{\text{3-SAT}}$, Lemma 45, proved using Claim 25, shows lower bounds on $\mathcal{S}(f(A\mathbf{z}))$, for all $A \in \mathrm{GL}(|\mathbf{z}|, \mathbb{F})$. Proposition 46 proves α-gap-ETsparse is NP-hard using Lemma 45 and has the same proof as Proposition 44.

▶ **Lemma 45.** *Let $\psi \in \overline{\text{3-SAT}}$, f be as defined in* (4) *corresponding to ψ and $A \in \mathrm{GL}(|\mathbf{z}|, \mathbb{F})$.*

1. *If $A(x_0)$ or $A(y_0)$ is a linear form in at least 2 variables, then $\mathcal{S}(f(A\mathbf{z})) \geq d_1 + 1$.*
2. *If A is not as in item 1 and if $A(x_j)$ is a linear form in at least 2 variables for some $j \in [n]$, then $\mathcal{S}(f(A\mathbf{z})) \geq d_2 + 1$.*
3. *If A is not as in items 1 and 2 and $\mathcal{S}(A(y_j + x_j)) \geq 3$ or $\mathcal{S}(A(y_j - x_j)) \geq 3$ for some $j \in [n]$, then $\mathcal{S}(f(A\mathbf{z})) \geq \frac{d_3^2+3d_3+2}{2}$.*
4. *If A is not of the form described in the previous three cases, then $\mathcal{S}(f(A\mathbf{z})) \geq (d_4 + 1)^3$.*

▶ **Proposition 46.** *Let $\mathrm{char}(\mathbb{F}) = 0$. Then, α-gap-ETsparse is NP-hard for s-sparse homogeneous polynomial inputs over $\mathbb{F}$ and $\alpha = s^{\frac{1}{3}-\epsilon}$, where $\epsilon \in (0, \frac{1}{3})$ is an arbitrary constant.*

▶ Remark 47. The proof of Lemma 45 uses Claim 25, which works over $\mathrm{char}(\mathbb{F}) = 0$ fields. We believe that Claim 25 can be modified for finite characteristic fields, using which the argument in this section can be extended over such fields.

5 NP-hardness of ETsupport

In this section, we prove Theorem 16 for characteristic 0 fields. The full version contains the proofs of the lemmas and the reduction for the finite characteristic case.

Proof sketch. We map ψ, a 3-CNF, to a polynomial f, which is the sum of degree separated polynomials with at least one polynomial of support $\sigma+1$ (σ is a constant integer) and the rest of support σ. As the summands are degree separated, $\text{Supp}(f) = \sigma + 1$ and for any invertible linear transform A, $\text{Supp}(A(f))$ is the maximum support size over all the transformed summands. Claim 26 is used to show $\psi \in$ 3-SAT iff there exists an invertible linear transform A, such that $\text{Supp}(A(f)) \leq \sigma$. Thus, the reduction also holds for $(\sigma+1)$-to-σ ETsupport.

5.1 Construction of f and σ

Let $\sigma \geq 6$ be an even integer constant and ψ be as denoted in Section 3.1. Assume $n \geq \sigma + 4$ and that in the first clause of ψ all the variables are complemented.[19] Let $\mathbf{x} := \{x_1, \dots, x_n\}$, $\mathbf{y} := \{y_1, \dots, y_n\}$ and $\mathbf{z} := \{z_1, \dots, z_{\sigma-5}\}$ and $\mathbf{w} := \mathbf{x} \sqcup \mathbf{y} \sqcup \mathbf{z}$. Consider the polynomials:

- First, introduce $\binom{n+\sigma-5}{\sigma}$ many monomials defined by the set
$$P := \{(w_{i_1} \cdots w_{i_\sigma})^\star \mid w_{i_1}, \dots, w_{i_\sigma} \in \mathbf{z} \sqcup \mathbf{x} \ \text{ and are pairwise distinct}\}.$$
- Then, introduce $\binom{n}{\frac{\sigma}{2}}$ many monomials defined by the set
$$Q := \{((x_{i_1}y_{i_1}) \cdots (x_{i_{\frac{\sigma}{2}}} y_{i_{\frac{\sigma}{2}}}))^\star \mid i_1, \dots, i_{\frac{\sigma}{2}} \in [n] \ \text{ and are pairwise distinct}\}.$$
- Let $R := \{R_k(\mathbf{w}) \mid k \in [m]\}$, where $R_k(\mathbf{w})$ is defined corresponding to the k^{th} clause as:
$$R_k(\mathbf{w}) := (\prod_{j \in C_k} (y_j - a_{k,j}x_j))^2 (z_1 \cdots z_{\sigma-5})^\star.$$

Define $f(\mathbf{w}) := \sum_{g \in P} g(\mathbf{w}) + \sum_{h \in Q} h(\mathbf{w}) + \sum_{k=1}^m R_k(\mathbf{w})$. The degrees, denoted by $\star$, are of form $\sigma + i$ where $i \in [N]$ and $N = \binom{n+\sigma-5}{\sigma} + \binom{n}{\sigma/2} + m$ to ensure all polynomials in $P \sqcup Q \sqcup R$ are degree separated with degrees $\geq \sigma + 1$. Based on this, Observation 48 holds.

▶ **Observation 48.** $\mathcal{S}(f(\mathbf{w})) = O(n^\sigma + m)$ *and* $\text{Supp}(f(\mathbf{w})) = \sigma + 1$.

5.2 The forward direction

Proposition 49 shows how a satisfying assignment for ψ implies the existence of an invertible A, such that $\text{Supp}(f(A\mathbf{w})) = \sigma$ by constructing A from the satisfying assignment.

▶ **Proposition 49.** *Let* $\psi \in$ 3-SAT *with* $(u_1, \dots, u_n) \in \{0,1\}^n$ *a satisfying assignment. Then,* $\text{Supp}(f(A\mathbf{w})) = \sigma$*, where the transform* A *is defined as*

$$A : z_j \mapsto z_j, \ x_i \mapsto x_i, \ y_i \mapsto y_i + (1-u_i)x_i \quad i \in [n], j \in [\sigma - 5]. \tag{7}$$

[19] To achieve $n \geq \sigma + 4$, add fresh variables and clauses in these variables to ψ. To ensure that the first clause contains only complemented variables, every uncomplemented variable x in the first clause is replaced by $\neg x$ followed by complementing each occurrence of x in the remaining clauses of ψ.

Proof. As all polynomials in $P \sqcup Q \sqcup R$ are degree separated, analysing the action of A on individual polynomials suffices. For $g \in P$, $g(A\mathbf{w}) = g$. On each monomial of Q, A acts as:

$$A : ((x_{i_1} y_{i_1}) \cdots (x_{i_{\frac{s}{2}}} y_{i_{\frac{\sigma}{2}}}))^{\star} \mapsto ((x_{i_1})(y_{i_1} + (1 - u_{i_1})x_{i_1}) \cdots (x_{i_{\frac{\sigma}{2}}})(y_{i_{\frac{\sigma}{2}}} + (1 - u_{i_{\frac{s}{2}}})x_{i_{\frac{\sigma}{2}}}))^{\star}.$$

Under A, each monomial of Q has support σ by Claim 26. For $k \in [m]$, A acts on $R_k(\mathbf{w})$ as:

$$A : (\prod_{j \in C_k} (y_j - a_{k,j} x_j))^2 \cdot (z_1 \cdots z_{\sigma-5})^{\star} \mapsto (\prod_{j \in C_k} (y_j + (1 - a_{k,j} - u_j)x_j)^2 \cdot (z_1 \cdots z_{\sigma-5})^{\star}.$$

If $a_{k,j} \neq u_j$, then $a_{k,j} = 1 - u_j$. Since ψ is satisfiable, therefore for all $k \in [m]$, $a_{k,j} \neq u_j$ for some $j \in C_k$. Hence, $\mathrm{Supp}(R_k(A\mathbf{w})) \leq (\sigma - 5) + 5 = \sigma$ for all $k \in [m]$. ◀

5.3 The reverse direction

Now, we show that if $\mathrm{Supp}(f(A\mathbf{w})) \leq \sigma$ for $A \in \mathrm{GL}(|\mathbf{w}|, \mathbb{F})$, then a satisfying assignment can be obtained for ψ. Lemmas 50 and 51, proved using Claim 26, together show that A is as:

$$A : z_j \mapsto z_j,\ x_i \mapsto x_i,\ y_i \mapsto y_i + c_i x_i \quad c_i \in \mathbb{F},\ j \in [\sigma - 5], i \in [n]$$

without loss of generality.[20] Proposition 52 constructs a satisfying assignment for ψ from A.

▶ **Lemma 50.** *If* $\mathrm{Supp}(f(A\mathbf{w})) \leq \sigma$*, then* $\forall w \in \mathbf{z} \sqcup \mathbf{x}$*,* $A(w) = W$*, for scaled variable* $W \in \mathbf{w}$*.*

▶ **Lemma 51.** *If* $\mathrm{Supp}(f(A\mathbf{w})) \leq \sigma$*, then* $A(x_i) = X_i$ *and* $A(y_i) = Y_i + c_i X_i$*, for scaled variables* $Y_i, X_i \in \mathbf{w}$*.*

▶ **Proposition 52.** *A satisfying assignment* $\mathbf{u}$ *for* ψ *can be constructed from* A*.*

Proof. The action of A on R_k, where $k \in [m]$, is:

$$(\prod_{j \in C_k} (y_j - a_{k,j} x_j))^2 \cdot (z_1 \cdots z_{\sigma-5})^{\star} \mapsto (\prod_{j \in C_k} (y_j + (c_j - a_{k,j})x_j))^2 \cdot (z_1 \cdots z_{\sigma-5})^{\star}.$$

Thus, $\mathrm{Supp}(R_k(A\mathbf{w})) \leq \sigma$ iff for some $j \in C_k$, $c_j = a_{k,j}$. By assumption $\mathrm{Supp}(R_k(A\mathbf{w})) \leq \sigma$ for all $k \in [m]$. Hence, for each $R_k(\mathbf{w})$, there exists $j \in C_k$ such that $c_j \in \{0, 1\}$. Construct $\mathbf{u} \in \{0, 1\}^n$ by setting $u_j := 1 - c_j$, for appropriate $j \in C_k$ and the remaining u_i's to arbitrary values in $\{0, 1\}$. From the definition of $\mathbf{u}$, it follows that for all $k \in [m]$, there exists $j \in C_k$ such that $u_j \neq a_{k,j}$. As k is arbitrary, all clauses are satisfied. ◀

6 Conclusion

In this work, we show that ET for sparse polynomials is NP-hard. Particularly, we show the NP-hardness of MCSP for orbits of homogeneous sparse polynomials (a dense subclass of hom-$\Sigma\Pi\Sigma$ circuits) over characteristic 0 fields. We also define a gap version of ET for sparse polynomials and show it is NP-hard, which implies the NP-hardness of $s^{\frac{1}{3}-\epsilon}$-factor approximation of the sparse-orbit complexity of s-sparse polynomials. Lastly, we show that ET for constant-support polynomials is NP-hard. In all three cases, we reduce 3-SAT to the respective problems. We end by listing some problems whose solutions we do not know:

[20] as permutation and non-zero scaling of variables do not affect the support.

1. **Hardness of ETsparse for constant degree polynomials:** In the reduction of Theorem 2, can the degree of the output polynomial be made constant? Currently, the degree is polynomial in the number of clauses and variables.
2. **Improving the gap in Theorem 8:** Can α-gap-ETsparse be shown NP-hard for $\alpha = s^{1-\epsilon}$, where the input polynomial has sparsity s and $\epsilon > 0$ is an arbitrary constant?
3. **Hardness of ETsupport for $\sigma = 2$:** Is checking if a given polynomial is in the orbit of a support-2 polynomial NP-hard? Theorem 16 shows ETsupport is NP-hard for $\sigma \geq 6$.
4. **Hardness of MCSP for hom-$\Sigma\Pi\Sigma$ circuits:** Is MCSP for hom-$\Sigma\Pi\Sigma$ circuits NP-hard?

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 Manindra Agrawal and Nitin Saxena. Automorphisms of finite rings and applications to complexity of problems. In *Proceedings of the 22nd Annual Conference on Theoretical Aspects of Computer Science*, STACS'05, pages 1–17, Berlin, Heidelberg, 2005. Springer-Verlag. doi:10.1007/978-3-540-31856-9_1.

3 Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. *Inf. Comput.*, 256:2–8, 2017. Conference version appeared in the proceedings of MFCS 2014. doi:10.1016/J.IC.2017.04.004.

4 Eric Allender, Lisa Hellerstein, Paul McCabe, Toniann Pitassi, and Michael E. Saks. Minimizing DNF Formulas and $\mathrm{AC}^0{}_\mathrm{d}$ Circuits Given a Truth Table. In *21st Annual IEEE Conference on Computational Complexity (CCC 2006), 16-20 July 2006, Prague, Czech Republic*, pages 237–251. IEEE Computer Society, 2006. doi:10.1109/CCC.2006.27.

5 Prashanth Amireddy, Ankit Garg, Neeraj Kayal, Chandan Saha, and Bhargav Thankey. Low-depth arithmetic circuit lower bounds: Bypassing set-multilinearization. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pages 12:1–12:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.ICALP.2023.12.

6 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998. Conference version appeared in the proceedings of FOCS 1992. doi:10.1145/278298.278306.

7 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998. Conference version appeared in the proceedings of FOCS 1992. doi:10.1145/273865.273901.

8 Omkar Baraskar, Agrim Dewan, and Chandan Saha. Testing equivalence to design polynomials. In Olaf Beyersdorff, Mamadou Moustapha Kanté, Orna Kupferman, and Daniel Lokshtanov, editors, *41st International Symposium on Theoretical Aspects of Computer Science, STACS 2024, March 12-14, 2024, Clermont-Ferrand, France*, volume 289 of *LIPIcs*, pages 9:1–9:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.STACS.2024.9.

9 Amos Beimel, Francesco Bergadano, Nader H. Bshouty, Eyal Kushilevitz, and Stefano Varricchio. Learning functions represented as multiplicity automata. *J. ACM*, 47(3):506–530, 2000. Conference version appeared in the proceedings of FOCS 1996. doi:10.1145/337244.337257.

10 Michael Ben-Or and Prasoon Tiwari. A Deterministic Algorithm for Sparse Multivariate Polynominal Interpolation (Extended Abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 301–309. ACM, 1988. doi:10.1145/62212.62241.

11 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. Conference version appeared in the proceedings of FOCS 2018. doi:10.1145/3365667.

12 Markus Bläser and Gorav Jindal. A new deterministic algorithm for sparse multivariate polynomial interpolation. In Katsusuke Nabeshima, Kosaku Nagasaka, Franz Winkler, and Ágnes Szántó, editors, *International Symposium on Symbolic and Algebraic Computation, ISSAC '14, Kobe, Japan, July 23-25, 2014*, pages 51–58. ACM, 2014. doi:10.1145/2608628.2608648.

13 Lenore Blum, Mike Shub, and Steve Smale. On a Theory of Computation and Complexity over the Real Numbers: NP-completeness, Recursive Functions and Universal Machines. *Bulletin of the American Mathematical Society*, 21(1):1–46, 1989. doi:10.1090/S0273-0979-1989-15750-9.

14 Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning Algorithms from Natural Proofs. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 10:1–10:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.CCC.2016.10.

15 Suryajith Chillara, Coral Grichener, and Amir Shpilka. On hardness of testing equivalence to sparse polynomials under shifts. In Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté, editors, *40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023, March 7-9, 2023, Hamburg, Germany*, volume 254 of *LIPIcs*, pages 22:1–22:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.STACS.2023.22.

16 Richard A. DeMillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Inf. Process. Lett.*, 7(4):193–195, 1978. doi:10.1016/0020-0190(78)90067-4.

17 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007. Conference version appeared in the proceedings of STOC 2006. doi:10.1145/1236457.1236459.

18 Michael A. Forbes. Some concrete questions on the border complexity of polynomials, 2016. URL: https://www.youtube.com/watch?v=1HMogQIHT6Q.

19 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, 26-29 October, 2013, Berkeley, CA, USA*, pages 243–252. IEEE Computer Society, 2013. doi:10.1109/FOCS.2013.34.

20 Dima Grigoriev and Marek Karpinski. A Zero-Test and an Interpolation Algorithm for the Shifted Sparse Polynominals. In Gérard D. Cohen, Teo Mora, and Oscar Moreno, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 10th International Symposium, AAECC-10, San Juan de Puerto Rico, Puerto Rico, May 10-14, 1993, Proceedings*, volume 673 of *Lecture Notes in Computer Science*, pages 162–169. Springer, 1993. doi:10.1007/3-540-56686-4_41.

21 Dima Grigoriev, Marek Karpinski, and Michael F. Singer. Fast Parallel Algorithms for Sparse Multivariate Polynomial Interpolation over Finite Fields. *SIAM J. Comput.*, 19(6):1059–1063, 1990. doi:10.1137/0219073.

22 Dima Grigoriev and Alexander A. Razborov. Exponential Complexity Lower Bounds for Depth 3 Arithmetic Circuits in Algebras of Functions Over Finite Fields. In *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*, pages 269–278. IEEE Computer Society, 1998. doi:10.1109/SFCS.1998.743456.

23 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016. Conference version appeared in the proceedings of FOCS 2013. doi:10.1137/140957123.

24 Nikhil Gupta, Chandan Saha, and Bhargav Thankey. Equivalence Test for Read-Once Arithmetic Formulas. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 4205–4272. SIAM, 2023. doi:10.1137/1.9781611977554.ch162.

25 Thomas R. Hancock, Tao Jiang, Ming Li, and John Tromp. Lower Bounds on Learning Decision Lists and Trees. *Inf. Comput.*, 126(2):114–122, 1996. doi:10.1006/INCO.1996.0040.

26 Johan Håstad. Tensor Rank is NP-Complete. *J. Algorithms*, 11(4):644–654, 1990. Conference version appeared in the proceedings of ICALP 1989. doi:10.1016/0196-6774(90)90014-6.

27 Shuichi Hirahara. Non-Black-Box Worst-Case to Average-Case Reductions within NP. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 247–258. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00032.

28 Shuichi Hirahara. NP-Hardness of Learning Programs and Partial MCSP. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 – November 3, 2022*, pages 968–979. IEEE, 2022. doi:10.1109/FOCS54457.2022.00095.

29 Shuichi Hirahara, Igor C. Oliveira, and Rahul Santhanam. NP-hardness of Minimum Circuit Size Problem for OR-AND-MOD Circuits. In Rocco A. Servedio, editor, *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, volume 102 of *LIPIcs*, pages 5:1–5:31. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICS.CCC.2018.5.

30 Rahul Ilango. Constant Depth Formula and Partial Function Versions of MCSP are Hard. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 424–433. IEEE, 2020. doi:10.1109/FOCS46700.2020.00047.

31 Rahul Ilango. The Minimum Formula Size Problem is (ETH) Hard. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 427–432. IEEE, 2021. doi:10.1109/FOCS52979.2021.00050.

32 Rahul Ilango, Bruno Loff, and Igor C. Oliveira. NP-Hardness of Circuit Minimization for Multi-Output Functions. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 22:1–22:36. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.CCC.2020.22.

33 Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E Requires Exponential Circuits: Derandomizing the XOR Lemma. In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 220–229. ACM, 1997. doi:10.1145/258533.258590.

34 Valentine Kabanets and Jin-yi Cai. Circuit minimization problem. In F. Frances Yao and Eugene M. Luks, editors, *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 73–79. ACM, 2000. doi:10.1145/335305.335314.

35 Erich L. Kaltofen and Barry M. Trager. Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *J. Symb. Comput.*, 9(3):301–320, 1990. Conference version appeared in the proceedings of FOCS 1988. doi:10.1016/S0747-7171(08)80015-6.

36 Neeraj Kayal. Efficient algorithms for some special cases of the polynomial equivalence problem. In Dana Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1409–1421. SIAM, 2011. doi:10.1137/1.9781611973082.108.

37 Neeraj Kayal. Affine projections of polynomials: extended abstract. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19–22, 2012*, pages 643–662. ACM, 2012. doi:10.1145/2213977.2214036.

38 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. Conference version appeared in the proceedings of STACS 2016. doi:10.1145/3369928.

39 Neeraj Kayal, Vineet Nair, Chandan Saha, and Sébastien Tavenas. Reconstruction of full rank algebraic branching programs. *ACM Trans. Comput. Theory*, 11(1):2:1–2:56, 2019. Conference version appeared in the proceedings of CCC 2017. doi:10.1145/3282427.

40 Adam R. Klivans and Amir Shpilka. Learning restricted models of arithmetic circuits. *Theory Comput.*, 2(10):185–206, 2006. Conference version appeared in the proceedings of COLT 2003. doi:10.4086/TOC.2006.V002A010.

41 Adam R. Klivans and Daniel A. Spielman. Randomness efficient identity testing of multivariate polynomials. In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 216–223. ACM, 2001. doi:10.1145/380752.380801.

42 Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial Lower Bounds Against Low-Depth Algebraic Circuits. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 804–814. IEEE, 2021. doi:10.1109/FOCS52979.2021.00083.

43 Richard J. Lipton and Nisheeth K. Vishnoi. Deterministic identity testing for multivariate polynomials. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 12-14, 2003, Baltimore, Maryland, USA*, pages 756–760. ACM/SIAM, 2003. URL: http://dl.acm.org/citation.cfm?id=644108.644233.

44 W. J. Masek. Some NP-complete set covering problems. *Unpublished Manuscript*, 1979.

45 Dori Medini and Amir Shpilka. Hitting sets and reconstruction for dense orbits in vp_{e} and $\Sigma\Pi\Sigma$ circuits. In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPIcs*, pages 19:1–19:27. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.CCC.2021.19.

46 Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, volume 1070 of *Lecture Notes in Computer Science*, pages 33–48. Springer, 1996. doi:10.1007/3-540-68339-9_4.

47 Ján Pich and Rahul Santhanam. Why are Proof Complexity Lower Bounds Hard? In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 1305–1324. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00080.

48 Alexander A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Math. Notes*, 41:333–338, 1987. doi:10.1007/BF01137685.

49 Daniel S. Roche. What Can (and Can't) we Do with Sparse Polynomials? In Manuel Kauers, Alexey Ovchinnikov, and Éric Schost, editors, *Proceedings of the 2018 ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2018, New York, NY, USA, July 16-19, 2018*, pages 25–30. ACM, 2018. doi:10.1145/3208976.3209027.

50 Chandan Saha and Bhargav Thankey. Hitting sets for orbits of circuit classes and polynomial families. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pages 50:1–50:26. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.APPROX/RANDOM.2021.50.

51 Nitin Saxena. *Morphisms of rings and applications to complexity*. PhD thesis, Indian Institute of Technology, Kanpur, 2006. URL: https://www.cse.iitk.ac.in/users/manindra/Students/thesis_saxena.pdf.

52 Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980. doi:10.1145/322217.322225.

53 Yaroslav Shitov. How hard is the tensor rank?, 2016. arXiv:1611.01559.

54 Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Inf. Comput.*, 240:2–11, 2015. Conference version appeared in the proceedings of MFCS 2013. doi:10.1016/J.IC.2014.09.004.

55 Thomas Thierauf. The isomorphism problem for read-once branching programs and arithmetic circuits. *Chic. J. Theor. Comput. Sci.*, 1998, 1998. URL: http://cjtcs.cs.uchicago.edu/articles/1998/1/contents.html.

56 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.

57 Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation, Marseille, France, June 1979, Proceedings*, pages 216–226, 1979. doi:10.1007/3-540-09519-5_73.