On testing isomorphism of polynomials, algebras, and multilinear forms

Anonymous author
Anonymous affiliation
Anonymous author
Anonymous affiliation
Anonymous author
Anonymous affiliation

Abstract

We study the problems of testing isomorphism of polynomials, algebras, and multilinear forms. Our first set of results consists of average-case algorithms for these problems. For example, we devise an algorithm that takes a random cubic form $f \in \mathbb{F}_q[x_1, \ldots, x_n]$ and an arbitrary cubic form g, and decides whether f and g are isomorphic in time $q^{O(n)}$ for most f. Such an average-case setting is of practical value, as it has been studied in multivariate cryptography since the 1990s. Our second result concerns the complexity of testing equivalence of alternating trilinear forms. This problem is of interest both in mathematics and in cryptography. We show that this problem is polynomial-time equivalent to testing equivalence of symmetric trilinear forms, therefore almost equivalent to testing isomorphism of cubic forms.

2012 ACM Subject Classification Computing methodologies → Algebraic algorithms; Computing methodologies → Combinatorial algorithms

Keywords and phrases polynomial isomorphism, trilinear form equivalence, algebra isomorphism, average-case algorithms, tensor isomorphism complete, symmetric and alternating bilinear maps

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

In this paper, we study isomorphism testing problems for polynomials, algebras, and multilinear forms. Our first set of results is algorithmic, namely presenting average-case algorithms for these problems (Section 1.1). Our second result is complexity-theoretic, concerning the problems of testing equivalence of symmetric and alternating trilinear forms (Section 1.2).

1.1 Average-case algorithms for polynomial isomorphism and more

The polynomial isomorphism problem. Let $\mathbb{F}$ be a field, and let $X = \{x_1, \ldots, x_n\}$ be a set of variables. Let $\mathrm{GL}(n, \mathbb{F})$ be the general linear group consisting of $n \times n$ invertible matrices over $\mathbb{F}$. A natural group action of $A = (a_{i,j}) \in \mathrm{GL}(n, \mathbb{F})$ on the polynomial ring $\mathbb{F}[X]$ sends $f(x_1, \ldots, x_n)$ to $f \circ A := f(\sum_{j=1}^{n} a_{1,j}x_j, \ldots, \sum_{j=1}^{n} a_{n,j}x_j)$. The *polynomial isomorphism problem* (PI) asks, given $f, g \in \mathbb{F}[X]$, whether there exists $A \in \mathrm{GL}(n, \mathbb{F})$ such that $f = g \circ A$. In the literature, this problem was also called the polynomial equivalence problem [1].

An important subcase of PI is when the input polynomials are required to be homogeneous of degree d. In this case, this problem is referred to as the homogeneous polynomial isomorphism problem, denoted as d-HPI. Homogeneous degree-3 (resp. degree-2) polynomials are also known as cubic (resp. quadratic) forms.

In this article, we assume that a polynomial is represented in algorithms by its list of coefficients of the monomials, though other representations like algebraic circuits are also possible in this context [13]. Furthermore, we shall mostly restrict our attention to the case when the polynomial degrees are constant.

Motivations to study polynomial isomorphism. The polynomial isomorphism problem has been studied in both multivariate cryptography and computational complexity. In 1996, inspired by the celebrated zero-knowledge protocol for graph isomorphism [9], Patarin proposed to use PI as the security basis of authentication and signature protocols [17]. This lead to a series of works on practical algorithms for PI; see [3, 4, 12] and references therein. In the early 2000s, Agrawal, Kayal and Saxena studied PI from the computational complexity perspective. They related PI with graph isomorphism and algebra isomorphism [1, 2], and studied some special instances of PI [13] as well as several related algorithmic tasks [18].

Despite these works, little progress has been made on algorithms *with rigorous analysis* for the *general* PI. That is, the algorithms from multivariate cryptography [4] either are heuristic, or need unproven assumptions. The algorithm from [13] could only handle the case when f and g are isomorphic to a common *multilinear* polynomial h. While these previous works contain several nice ideas and insights, and their implementations show practical improvements, they are nonetheless heuristic in nature, and rigorously analyzing them seems difficult. Indeed, if any of these algorithms had worst-case analysis matching their heuristic performance, it would lead to significant progress on the long-open Group Isomorphism problem (see, e.g., [11, 15]).

Our result on polynomial isomorphism. Our first result is an average-case algorithm with rigorous analysis for PI over a finite field $\mathbb{F}_q$. As far as we know, this is the first non-trivial algorithm with rigorous analysis for PI over finite fields. (The natural brute-force algorithm, namely enumerating all invertible matrices, runs in time $q^{n^2} \cdot \mathrm{poly}(n, \log q)$.) Furthermore, the average-case setting is quite natural, as it is precisely the one studied multivariate cryptography. We shall elaborate on this further after stating our result.

To state the result, let us define what a random polynomial means in this setting. Since we represent polynomials by their lists of coefficients, a random polynomial of degree d is naturally the one whose coefficients of the monomials of degree $\leq d$ are independently randomly drawn from $\mathbb{F}_q$. We also consider the homogeneous setting where only monomials of degree $= d$ are of interest.

▶ **Theorem 1.** *Let $d \geq 3$ be a constant. Let $f \in \mathbb{F}_q[x_1, \ldots, x_n]$ be a random (resp. homogeneous) polynomial of degree $\leq d$ (resp. $= d$). There exists an algorithm in time $q^{O(n)}$ that decides whether f is isomorphic to an arbitrary g, for all but at most $\frac{1}{q^{\Omega(n)}}$ fraction of f.*

Furthermore, if f and g are isomorphic, then this algorithm also computes an invertible matrix A which sends f to g.

Let us briefly indicate the use of this average-case setting in multivariate cryptography. In the authentication scheme described in [17], the public key consists of two polynomials $f, g \in \mathbb{F}_q[x_1, \ldots, x_n]$, where f is a random polynomial, and g is obtained by applying a random invertible matrix to f. Then f and g are public keys, and any isomorphism from f to g can serve as the private key. Therefore, the algorithm in Theorem 1 can be used to recover a private key for most f.

Adapting the algorithm strategy to more isomorphism problems. In [1, 2], the algebra isomorphism problem (AI) was studied and shown to be (almost) polynomial-time equivalent to PI. In [11], many more problems are demonstrated to be polynomial-time equivalent to PI, including the trilinear form equivalence problem (TFE). In these reductions, due to the blow-up of the parameters, the $q^{O(n)}$-time algorithm in Theorem 1 does not translate to moderately exponential-time, average-case algorithms for these problems. The algorithm design idea, however, does translate to give moderately exponential-time, average-case algorithms for AI and TFE. This will be shown in Section 3.2.

1.2 Complexity of symmetric and alternating trilinear form equivalence

From cubic forms to symmetric and alternating trilinear forms. In the context of polynomial isomorphism, cubic forms are of particular interest. In complexity theory, it was shown that d-HPI reduces to cubic form isomorphism over fields with dth roots of unity [1, 2]. In multivariate cryptography, cubic form isomorphism also received special attention, since using higher degree forms results in less efficiency in the cryptographic protocols.

Just as quadratic forms are closely related with symmetric bilinear forms, cubic forms are closely related with symmetric trilinear forms. Let $\mathbb{F}$ be a field of characteristic not 2 or 3, and let $f = \sum_{1 \leq i \leq j \leq k \leq n} a_{i,j,k} x_i x_j x_k \in \mathbb{F}[x_1, \ldots, x_n]$ be a cubic form. For any $i, j, k \in [n]$, let $1 \leq i' \leq j' \leq k' \leq n$ be the result of sorting i, j, k in the increasing order, and set $a_{i,j,k} = a_{i',j',k'}$. Then we can define a symmetric[1] trilinear form $\phi_f : \mathbb{F}^n \times \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by $\phi(u, v, w) = \frac{1}{6} \sum_{i,j,k \in [n]} a_{i,j,k} u_i v_j w_k$. It can be seen easily that for any $v = (v_1, \ldots, v_n)^{\mathrm{t}} \in \mathbb{F}_q^n$, $f(v_1, \ldots, v_n) = \phi(v, v, v)$.

In the theory of bilinear forms, symmetric and skew-symmetric bilinear forms are two important special subclasses. For example, they are critical in the classifications of classical groups [19] and finite simple groups [20]. For trilinear forms, we also have skew-symmetric trilinear forms. In fact, to avoid some complications over fields of characteristics 2 or 3, we shall consider alternating trilinear forms which are closely related to skew-symmetric ones.

[1] That is, for any permutation $\sigma \in \mathrm{S}_3$, $\phi(u_1, u_2, u_3) = \phi(u_{\sigma(1)}, u_{\sigma(2)}, u_{\sigma(3)})$

We say that a trilinear form $\phi : \mathbb{F}^n \times \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ is *alternating*, if whenever two arguments of ϕ are equal, ϕ evaluates to zero. Note that this implies skew-symmetry, namely for any u_1, u_2, u_3 and any $\sigma \in \mathrm{S}_3$, $\phi(u_1, u_2, u_3) = \mathrm{sgn}(\sigma) \cdot \phi(u_{\sigma(1)}, u_{\sigma(2)}, u_{\sigma(3)})$. Over fields of characteristic zero or > 3, this is equivalent to skew-symmetry.

The trilinear form equivalence problem. Given a trilinear form $\phi : \mathbb{F}^n \times \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$, $A \in \mathrm{GL}(n, \mathbb{F})$ naturally acts on ϕ by sending it to $\phi \circ A := \phi(A^{-1}(u), A^{-1}(v), A^{-1}(w))$. The *trilinear form equivalence problem* then asks, given two trilinear forms $\phi, \psi : \mathbb{F}^n \times \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$, whether there exists $A \in \mathrm{GL}(n, \mathbb{F})$, such that $\phi = \psi \circ A$. Over fields of characteristic not 2 or 3, two cubic forms f and g are isomorphic if and only if ϕ_f and ϕ_g are equivalent, so cubic form isomorphism is polynomial-time equivalent to symmetric form equivalence over such fields. Note that for clarify, we reserve isomorphism for polynomials (and cubic forms), and equivalence for multilinear forms.

Motivations to study alternating trilinear form equivalence. Our main interest is to study the complexity of alternating trilinear form equivalence, with the following motivations.

The first motivation comes from cryptography. To store a symmetric trilinear form on $\mathbb{F}_q^n$, $\binom{n+2}{3}$ field elements are required. To store an alternating trilinear form on $\mathbb{F}_q^n$, $\binom{n}{3}$ field elements are needed. The difference between $\binom{n+2}{3}$ and $\binom{n}{3}$ could be significant for practical purposes. For example, when $n = 9$, $\binom{n+2}{3} = \binom{11}{3} = 165$, while $\binom{n}{3} = \binom{9}{3} = 84$. This means that in the authentication protocol of Patarin [17], using alternating trilinear forms instead of cubic forms for $n = 9$,[2] one saves almost one half in the public key size, which is an important saving in practice.

The second motivation originates from comparing symmetric and alternating bilinear forms. It is well-known that, *in the bilinear case*, the structure of alternating forms is simpler than that for symmetric ones [14]. Indeed, up to equivalence, an alternating bilinear form is completely determined by its rank over any field, while the classification of symmetric bilinear forms depends crucially on the underlying field. For example, recall that over $\mathbb{R}$, a symmetric form is determined by its "signature", so just the rank is not enough.

A third motivation is implied by the representation theory of the general linear groups; namely that alternating trilinear forms are the "last" natural case for $d = 3$. If we consider the action of $\mathrm{GL}(n, \mathbb{C})$ acting on d-tensors in $\mathbb{C}^n \otimes \mathbb{C}^n \otimes \cdots \otimes \mathbb{C}^n$ diagonally (that is, the same matrix acts on each tensor factor), it is a classical result [19] that the invariant subspaces of $(\mathbb{C}^n)^{\otimes d}$ under this action are completely determined by the irreducible representations of $\mathrm{GL}(n, \mathbb{C})$. When $d = 3$, there are only three such representations, which correspond precisely to: symmetric trilinear forms, Lie algebras, and alternating trilinear forms. From the complexity point of view, it was previously shown that isomorphism of symmetric trilinear forms [1, 2] and Lie algebras [11] are equivalent to algebra isomorphism. Here we show that the last case, isomorphism of alternating trilinear forms, is also equivalent the others.

The complexity of alternating trilinear form equivalence. Given the above discussion on the comparison between symmetric and alternating bilinear forms, one may wonder whether alternating trilinear form equivalence was easier than symmetric trilinear form equivalence. Interestingly, we show that this is not the case; rather, they are polynomial-time equivalent.

▶ **Theorem 2.** *The alternating trilinear form equivalence problem is polynomial-time equivalent to the symmetric trilinear form equivalence problem.*

[2] The parameters of the cryptosystem are q and n. When $q = 2$, $n = 9$ is not secure as it can be solved in practice [5]. So q needs to be large for $n = 9$ to be secure. Interestingly, according to [4, pp. 227], the parameters $q = 16$ and $n = 8$ seemed difficult for practical attacks via Gröbner basis.

1.3 Previous works

The relation between PI and AI. As mentioned in Section 1.1, the degree-d homogeneous polynomial isomorphism problem (d-HPI) was shown to be almost equivalent to the algebra isomorphism problem (AI) in [1, 2]. Here, almost refers to that for the reduction from d-HPI to AI in [1, 2], the underlying fields are required to contain a dth root of unity. When $d = 3$, this means that the characteristic of the underlying field p satisfies that $p = 2 \mod 3$ or $p = 0$, which amounts to half of the primes. In [11], another reduction from 3-HPI to AI was presented, which works for fields of characteristics not 2 or 3. The reduction from AI to 3-HPI in [2] works over any field.

The tensor isomorphism complete class. In [8, 11], polynomial-time equivalences are proved between isomorphism testing of many more mathematical structures, including tensors, matrix spaces, polynomial maps, and so on. These problems arise from many areas: besides multivariate cryptography and computational complexity, they appear in quantum information, machine learning, and computational group theory. This motivates the authors of [11] to define the tensor isomorphism complete class TI, which we recall here.

▶ **Definition 3** (The d-Tensor Isomorphism problem, and the complexity class TI)**.** d-Tensor Isomorphism *over a field* $\mathbb{F}$ *is the problem: given two* d*-way arrays* $\mathtt{A} = (a_{i_1,\dots,i_d})$ *and* $\mathtt{B} = (b_{i_1,\dots,i_d})$*, where* $i_k \in [n_k]$ *for* $k \in [d]$*, and* $a_{i_1,\dots,i_d}, b_{i_1,\dots,i_d} \in \mathbb{F}$*, decide whether there are* $P_k \in \mathrm{GL}(n_k, \mathbb{F})$ *for* $k \in [d]$*, such that for all* $i_1, \dots, i_d$*,*

$$a_{i_1,\dots,i_d} = \sum_{j_1,\dots,j_d} b_{j_1,\dots,j_d} (P_1)_{i_1,j_1} (P_2)_{i_2,j_2} \cdots (P_d)_{i_d,j_d}. \tag{1}$$

For any field $\mathbb{F}$*,* $\mathrm{TI}_{\mathbb{F}}$ *denotes the class of problems that are polynomial-time Turing (Cook) reducible to* d-Tensor Isomorphism *over* $\mathbb{F}$*, for some* d*. A problem is* $\mathrm{TI}_{\mathbb{F}}$*-complete, if it is in* $\mathrm{TI}_{\mathbb{F}}$*, and* d-Tensor Isomorphism *over* $\mathbb{F}$ *for any* d *reduces to this problem.*

When a problem is naturally defined, and is $\mathrm{TI}_{\mathbb{F}}$*-complete, over any* $\mathbb{F}$*, then we can simply write that it is* TI*-complete.*

Average-case algorithms for matrix space isometry. In [6, 15], motivated by testing isomorphism of p-groups (widely believed to be the hardest cases of Group Isomorphism, see e.g. [10]), the algorithmic problem alternating matrix space isometry was studied. That problem asks, given two linear spaces of alternating matrices $\mathcal{A}, \mathcal{B} \leq \Lambda(n, q)$, to decide whether there exists $T \in \mathrm{GL}(n, q)$, such that $\mathcal{A} = T^{\mathrm{t}}\mathcal{B}T = \{T^{\mathrm{t}}BT : B \in \mathcal{B}\}$. (See Section 2 for the definition of alternating matrices.) The main result of [6], improving upon the one in [15], is an average-case algorithm for this problem in time $q^{O(n+m)}$, where $m = \dim(\mathcal{A})$.

1.4 Remarks on the technical side

Techniques for proving Theorem 1. The algorithm for PI in Theorem 1 is based on the algorithmic idea from [6, 15]. However, to adapt that idea to the PI setting does meet several interesting conceptual and technical difficulties.

One conceptual difficulty is that for alternating matrix space isometry, there are actually two GL actions, one is by $\mathrm{GL}(n, q)$ as explicitly described above, and the other is by $\mathrm{GL}(m, q)$ performing the changes of bases of matrix spaces. The algorithm in [6] crucially uses that the $\mathrm{GL}(m, q)$ action is "independent" of the $\mathrm{GL}(n, q)$ action. For PI, there is only one $\mathrm{GL}(n, q)$-action acting on all the variables. Luckily, as shown in Section 3.1, there is still a natural way of applying the the basic idea from [6, 15].

One technical difficulty is that the analysis in [6] relies on properties of random alternating matrices, while for 3-HPI, the analysis relies on properties of random symmetric matrices. To adapt the proof strategy in [6] (based on [15]) to the symmetric setting is not difficult, but suggests some interesting differences between symmetric and alternating matrices (see the discussion after Claim 13).

Techniques for proving Theorem 2. By [8], the trilinear form equivalence problem is in TI, and so are the special cases symmetric and alternating trilinear form equivalence. The proof of Theorem 2 goes by showing that both symmetric and alternating trilinear form equivalence are TI-hard.

Technically, the basic proof strategy is to adapt a gadget construction, which originates from [8] and then is further used in [11]. To use that gadget in the trilinear form setting does require several non-trivial ideas. First, we identify the right TI-complete problem to start with, namely the alternating (resp. symmetric) matrix space isometry problem. Second, we need to arrange a 3-way array $\mathtt{A}$, representing a linear basis of an alternating (resp. symmetric) matrix spaces, into one representing an alternating trilinear form. This requires 3 copies of $\mathtt{A}$, assembled in an appropriate manner. Third, we need to add the gadget in three directions (instead of just two as in previous results). All these features were not present in [8, 11]. The correctness proof also requires certain tricky twist compared with those in [8] and [11].

2 Preliminaries

Notations. We collect the notations here, though some of them have appeared in Section 1. Let $\mathbb{F}$ be a field. Vectors in $\mathbb{F}^n$ are column vectors. Let e_i denote the ith standard basis vector of $\mathbb{F}^n$. Let $\mathrm{M}(\ell \times n, \mathbb{F})$ be the linear space of $\ell \times n$ matrices over $\mathbb{F}$, and set $\mathrm{M}(n, \mathbb{F}) := \mathrm{M}(n \times n, \mathbb{F})$. Let I_n denote the identity matrix of size n. For $A \in \mathrm{M}(n, \mathbb{F})$, A is *symmetric* if $A^{\mathrm{t}} = A$, and *alternating* if for every $v \in \mathbb{F}^n$, $v^{\mathrm{t}}Av = 0$. When $\mathbb{F}$ is of characteristic not 2, A is alternating if and only if A is skew-symmetric. Let $\mathrm{S}(n, \mathbb{F})$ be the linear space of $n \times n$ symmetric matrices over $\mathbb{F}$, and let $\Lambda(n, \mathbb{F})$ be the linear space of alternating matrices over $\mathbb{F}$. When $\mathbb{F} = \mathbb{F}_q$, we may write $\mathrm{M}(n, \mathbb{F}_q)$ as $\mathrm{M}(n, q)$. We use $\langle \cdot \rangle$ to denote the linear span.

3-way arrays. A 3-way array over a field $\mathbb{F}$ is an array with three indices whose elements are from $\mathbb{F}$. We use $\mathrm{M}(n_1 \times n_2 \times n_3, \mathbb{F})$ to denote the linear space of 3-way arrays of side lengths $n_1 \times n_2 \times n_3$ over $\mathbb{F}$.

Let $\mathtt{A} \in \mathrm{M}(\ell \times n \times m, \mathbb{F})$. For $k \in [m]$, the kth *frontal* slice of $\mathtt{A}$ is $(a_{i,j,k})_{i\in[\ell],j\in[n]} \in \mathrm{M}(\ell \times n, \mathbb{F})$. For $j \in [n]$, the jth *vertical* slice of $\mathtt{A}$ is $(a_{i,j,k})_{i\in[\ell],k\in[m]} \in \mathrm{M}(\ell \times m, \mathbb{F})$. For $i \in [\ell]$, the ith *horizontal* slice of $\mathtt{A}$ is $(a_{i,j,k})_{j\in[n],k\in[m]} \in \mathrm{M}(n \times m, \mathbb{F})$. We shall often think of $\mathtt{A}$ as a matrix tuple in $\mathrm{M}(\ell \times n, \mathbb{F})^m$ consisting of its frontal slices.

A natural action of $(P, Q, R) \in \mathrm{GL}(\ell, \mathbb{F}) \times \mathrm{GL}(n, \mathbb{F}) \times \mathrm{GL}(m, \mathbb{F})$ sends a 3-way array $\mathtt{A} \in \mathrm{M}(\ell \times n \times m, \mathbb{F})$ to $P^{\mathrm{t}}\mathtt{A}^R Q$, defined as in Equation 1.

Useful results. Let $\mathbf{A} = (A_1, \ldots, A_m), \mathbf{B} = (B_1, \ldots, B_m) \in \mathrm{M}(n, \mathbb{F})^m$. Given $T \in \mathrm{GL}(n, \mathbb{F})$, let $T^{\mathrm{t}}\mathbf{A}T = (T^{\mathrm{t}}A_1T, \ldots, T^{\mathrm{t}}A_mT)$. We say that $\mathbf{A}$ and $\mathbf{B}$ are *isometric*, if there exists $T \in \mathrm{GL}(n, \mathbb{F})$ such that $T^{\mathrm{t}}\mathbf{A}T = \mathbf{B}$. Let $\mathrm{Iso}(\mathbf{A}, \mathbf{B}) = \{T \in \mathrm{GL}(n, \mathbb{F}) : \mathbf{A} = T^{\mathrm{t}}\mathbf{B}T\}$, and set $\mathrm{Aut}(\mathbf{A}) := \mathrm{Iso}(\mathbf{A}, \mathbf{A})$. Clearly, $\mathrm{Aut}(\mathbf{A})$ is a subgroup of $\mathrm{GL}(n, q)$, and $\mathrm{Iso}(\mathbf{A}, \mathbf{B})$ is a coset of $\mathrm{Aut}(\mathbf{A})$.

▶ **Theorem 4** ([7, 12]). *Let* $\mathbf{A}, \mathbf{B} \in \mathrm{S}(n,q)^m$ *(resp.* $\Lambda(n,q)^m$*) for some odd* q*. There exists a* $\mathrm{poly}(n,m,q)$*-time deterministic algorithm which takes* $\mathbf{A}$ *and* $\mathbf{B}$ *as inputs and outputs* $\mathrm{Iso}(\mathbf{A},\mathbf{B})$*, specified by (if nonempty) a generating set of* $\mathrm{Aut}(\mathbf{A})$ *(by the algorithm in [7]) and a coset representative* $T \in \mathrm{Iso}(\mathbf{A},\mathbf{B})$ *(by the algorithm in [12]).*

3 Average-case algorithms for polynomial isomorphism and more

We shall present the algorithm for the cubic form isomorphism problem in detail in Section 3.1. We will state our results for problems like algebra isomorphism in Section 3.2.

3.1 Cubic form isomorphism over fields of odd order

Due to page constraints, we present the algorithm for cubic form isomorphism over fields of odd characteristic, as this algorithm already captures the essense of the idea, and cubic forms are most interesting from the PI perspective as mentioned in Section 1.2. A full proof of Theorem 1, which is a relatively minor extension of Proposition 5, is put in Appendix B.

▶ **Proposition 5.** *Let* $\mathbb{F}_q$ *be a finite field of odd order. Let* $X = \{x_1, \ldots, x_n\}$ *be a set of commutative variables. Let* $f \in \mathbb{F}_q[X]$ *be a random cubic form, and let* $g \in \mathbb{F}_q[X]$ *be an arbitrary cubic form. There exists a deterministic algorithm that decides whether* f *and* g *are isomorphic in time* $q^{O(n)}$*, for all but at most* $\frac{1}{q^{\Omega(n)}}$ *fraction of* f*.*

Proof. Let r be a constant to be determined later on, and suppose n is sufficiently larger than r. Our goal is to find $T \in \mathrm{GL}(n,q)$, such that $f = g \circ T$.

The algorithm consists of two main steps. Let us first give an overview of the two steps.

In the first step, we show that there exists a set of at most $q^{O(rn)}$-many $T_1 \in \mathrm{GL}(n,q)$, such that every $T \in \mathrm{GL}(n,q)$ can be written as T_1T_2, where T_2 is of the form

$$\begin{bmatrix} I_r & 0 \\ 0 & R \end{bmatrix}. \tag{2}$$

Furthermore, such T_1 can be enumerated in time $q^{O(rn)}$. We then set $g_1 = g \circ T_1$.

In the second step, we focus on searching for T_2 such that $f = g_1 \circ T_2$. The key observation is that those T_2 as in Equation 2 leave x_i, $i \in [r]$, invariant, and send x_j, $j \in [r+1, n]$, to a linear combination of x_k, $k \in [r+1, n]$. It follows that for any fixed $i \in [r]$, T_2 sends $\sum_{r+1 \le j \le k \le n} a_{i,j,k} x_i x_j x_k$ to a linear combination of $x_i x_j x_k$, $r+1 \le j \le k \le n$. We will use this observation to show that for a random f, the number of T_2 satisfying $f = g_1 \circ T_2$ is upper bounded by q^n with high probability. Furthermore, such T_2, if they exist, can be enumerated efficiently. This allows us to go over all possible T_2 and test if $f = g_1 \circ T_2$.

The first step. We show that there exist at most $q^{O(rn)}$-many $T_1 \in \mathrm{GL}(n,q)$, such that any $T \in \mathrm{GL}(n,q)$ can be written as T_1T_2 where T_2 is of the form as in Equation 2.

Recall that e_i is the ith standard basis vector. Let $E_r = \langle e_1, \ldots, e_r \rangle$, and let $F_r = \langle e_{r+1}, \ldots, e_n \rangle$. Suppose for $r \in [n]$, $T(e_i) = u_i$, and $T(F_r) = V \le \mathbb{F}_q^n$. Let T_1 be any matrix that satisfies $T_1(e_i) = u_i$, and $T_1(F_r) = V$. Let $T_2 = T_1^{-1}T$. Then T_2 satisfies that for $i \in [r]$, $T_2(e_i) = e_i$, and $T_2(F_r) = F_r$. In other words, T_2 is of the form in Equation 2.

We then need to show that these T_1 can be enumerated in time $q^{O(rn)}$.

Recall that T_1 is determined by the images of e_i, $i \in [r]$, and $F_r \le \mathbb{F}_q^n$. So we first enumerate matrices of the form $\begin{bmatrix} u_1 & \ldots & u_r & e_{r+1} & \ldots & e_n \end{bmatrix}$, where $u_i \in \mathbb{F}_q^n$ are linearly independent. We then need to enumerate the possible images of F_r. Let $U = \langle u_1, \ldots, u_r \rangle$. Then the image of F_r is a complement subspace of U. It is well-known that the number of

complement subspaces of a dimension-r space is q^{rn}. To enumerate all complement subspaces of U, first compute one complement subspace $V = \langle v_1, \ldots, v_{n-r} \rangle$. Then it is easy to verify that, when going over $A = (a_{i,j})_{i\in[r],j\in[n-r]} \in \mathrm{M}(r\times(n-r), q)$, $\langle v_j + \sum_{i\in[r]} a_{i,j}u_i : j \in [n-r]\rangle$ go over all complement subspaces of U. It follows that we can enumerate matrices T_1 of the form $\begin{bmatrix} u_1 & \ldots & u_r & v_1 + \sum_{i\in[r]} a_{i,1}u_i & \ldots & v_{n-r} + \sum_{i\in[r]} a_{i,n-r}u_i \end{bmatrix}$.

The second step. In Step 1, we computed a set of invertible matrices $\{T_1\} \subseteq \mathrm{GL}(n, q)$ such that every $T \in \mathrm{GL}(n, q)$ can be written as $T = T_1T_2$ where $T_2 = \begin{bmatrix} I_r & 0 \\ 0 & R \end{bmatrix}$. So we set $g_1 := g \circ T_1$ and focus on finding T_2 of the above form such that $f = g_1 \circ T_2$.

Suppose $f = \sum_{1\le i\le j\le k\le n} \alpha_{i,j,k}x_ix_jx_k$, and $g_1 = \sum_{1\le i\le j\le k\le n} \beta_{i,j,k}x_ix_jx_k$. For $i \in [r]$, define $f_i = \sum_{r+1\le j\le k\le n} \alpha_{i,j,k}x_ix_jx_k$. Similarly define $g_{1,i}$.

The key observation is that, due to the form of T_2, we have that $f_i = g_{1,i} \circ T_2$. This is because for $i \in [r]$, T_2 sends x_i to x_i, and for $j \in [r+1, n]$, T_2 sends x_j to a linear combination of x_k, $k \in [r + 1, n]$.

Let $\ell = n - r$. We then rename the variable x_{r+i}, $i \in [\ell]$ as y_i. Let $Y = \{y_1, \ldots, y_\ell\}$. Then from f, we define r quadratic forms in Y,

$$\forall i \in [r], c_i = \sum_{1\le j\le k\le \ell} \alpha'_{i,j,k}y_jy_k, \text{ where } \alpha'_{i,j,k} = \alpha_{i,r+j,r+k}. \tag{3}$$

Correspondingly, we define r quadratic forms $d_i = \sum_{1\le j\le k\le \ell} \beta'_{i,j,k}y_jy_k$, $i \in [r]$, from g_1.

Our task now is to search for the $R \in \mathrm{GL}(\ell, q)$ such that for every $i \in [r]$, $c_i = d_i \circ R$.

To do that, we adopt the classical representation of quadratic forms as symmetric matrices. Here we use the assumption that q is odd. Using the classical correspondence between quadratic forms and symmetric matrices, from c_i we construct

$$C_i = \begin{bmatrix} \alpha'_{i,1,1} & \frac{1}{2}\alpha'_{i,1,2} & \cdots & \frac{1}{2}\alpha'_{i,1,\ell} \\ \frac{1}{2}\alpha'_{i,1,2} & \alpha'_{i,2,2} & \cdots & \frac{1}{2}\alpha'_{i,2,\ell} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2}\alpha'_{i,1,\ell} & \frac{1}{2}\alpha'_{i,2,\ell} & \cdots & \alpha'_{i,\ell,\ell} \end{bmatrix} \in \mathrm{S}(\ell, q) \tag{4}$$

from c_i. Similarly define D_i from d_i. It is classical that $c_i = d_i \circ R$ if and only if $C_i = R^{\mathrm{t}}D_iR$.

Let $\mathbf{C} = (C_1, \ldots, C_r) \in \mathrm{S}(\ell, q)^r$, and $\mathbf{D} = (D_1, \ldots, D_r) \in \mathrm{S}(\ell, q)^r$. Note that by our assumption, $\mathbf{C}$ is a tuple of random symmetric matrices. Recall that $\mathrm{Aut}(\mathbf{C}) = \{R \in \mathrm{GL}(n - r, \mathbb{F}) : R^{\mathrm{t}}\mathbf{C}R = \mathbf{C}\}$, and $\mathrm{Iso}(\mathbf{C}, \mathbf{D}) = \{R \in \mathrm{GL}(\ell, \mathbb{F}) : \mathbf{C} = R^{\mathrm{t}}\mathbf{D}R\}$. Clearly, $\mathrm{Iso}(\mathbf{C}, \mathbf{D})$ is a (possibly empty) coset of $\mathrm{Aut}(\mathbf{C})$. So when $\mathrm{Iso}(\mathbf{C}, \mathbf{D})$ is non-empty, $|\mathrm{Iso}(\mathbf{C}, \mathbf{D})| = |\mathrm{Aut}(\mathbf{C})|$. Our main technical lemma is the following, obtained by adapting certain results in [6, 15] to the symmetric matrix setting. Its proof can be found in Appendix A.

▶ **Lemma 6.** *Let $\mathbf{C} = (C_1, \ldots, C_8) \in \mathrm{S}(\ell, q)^8$ be a random symmetric matrix tuple. Then we have $|\mathrm{Aut}(\mathbf{C})| \le q^\ell$ for all but at most $\frac{1}{q^{\Omega(\ell)}}$ fraction of such $\mathbf{C}$.*

Given this lemma, we can use Theorem 4 to decide whether $\mathbf{C}$ and $\mathbf{D}$ are isometric, and if so, compute $\mathrm{Iso}(\mathbf{C}, \mathbf{D})$ represented as a coset in $\mathrm{GL}(\ell, q)$. By Lemma 6, for all but at most $\frac{1}{q^{\Omega(\ell)}}$ fraction of $\mathbf{C}$, $|\mathrm{Iso}(\mathbf{C}, \mathbf{D})| \le q^\ell \le q^n$. With $\mathrm{Iso}(\mathbf{C}, \mathbf{D})$ as a coset at hand, we can enumerate all elements in $\mathrm{Aut}(\mathbf{C})$ by the standard recursive closure algorithm [16] and therefore all elements in $\mathrm{Iso}(\mathbf{C}, \mathbf{D})$. We then either conclude that $|\mathrm{Iso}(\mathbf{C}, \mathbf{D})| > q^n$, or have all $\mathrm{Iso}(\mathbf{C}, \mathbf{D})$ at hand. In the former case we conclude that $\mathbf{C}$ does not satisfy the required generic condition. In the latter case, we enumerate $R \in \mathrm{Iso}(\mathbf{C}, \mathbf{D})$, and check whether $T_2 = \begin{bmatrix} I_r & 0 \\ 0 & R \end{bmatrix}$ is an isomorphism from f to g_1.

The algorithm outline. We now summarise the above steps in the following algorithm outline. In the following we assume that $n \gg 8$; otherwise we can use the brute-force algorithm.

Input Cubic forms $f, g \in \mathbb{F}_q[x_1, \ldots, x_n]$.

Output One of the following: (1) "f does not satisfy the generic condition"; (2) "f and g are not isomorphic"; (3) an isomorphism $T \in \mathrm{GL}(n, q)$ sending g to f.

Algorithm outline 1. Set $r = 8$, and $\ell = n - r$.

2. Compute $W = \{T_1\} \subseteq \mathrm{GL}(n, q)$ using the procedure described in Step 1.
   // `Every` $T \in \mathrm{GL}(n, q)$ `can be written as` T_1T_2 `where` T_2 `is of the form in Equation 2.`
3. For every $T_1 \in W$, do the following:
   a. $g_1 :\leftarrow g \circ T_1$.
   b. For $i \in [\ell]$, $y_i \leftarrow x_{r+i}$.
   c. For $i \in [r]$, let $C_i \in \mathrm{S}(\ell, q)$ be defined in Equation 4. Let $D_i \in \mathrm{S}(\ell, q)$ be defined from g_1 in the same way. Let $\mathbf{C} = (C_1, \ldots, C_r)$, and $\mathbf{D} = (D_1, \ldots, D_r)$.
   d. Use Theorem 4 to decide whether $\mathbf{C}$ and $\mathbf{D}$ are isometric. If not, break from the loop. If so, compute one isometry R.
   e. Use Theorem 4 to compute a generating set of $\mathrm{Aut}(\mathbf{C})$. Use the recursive closure algorithm to enumerate $\mathrm{Aut}(\mathbf{C})$. During the enumertion, if $|\mathrm{Aut}(\mathbf{C})| > q^\ell$, report "$f$ does not satisfy the generic condition." Otherwise, we have the whole $\mathrm{Aut}(\mathbf{C})$ at hand, which is of size $\leq q^\ell$.
   f. Given R from Line 3d and $\mathrm{Aut}(\mathbf{C})$ from Line 3e, the whole set $\mathrm{Iso}(\mathbf{C}, \mathbf{D})$ can be computed. For every $R \in \mathrm{Iso}(\mathbf{C}, \mathbf{D})$, check whether $T_2 = \begin{bmatrix} I_r & 0 \\ 0 & R \end{bmatrix}$ sends g_1 to f. If so, return $T = T_1T_2$ as an isomorphism sending g to f.
4. Return that "f and g are not isomorphic".

Correctness and timing analyses. The correctness of the algorithm relies on the simple fact that if f satisfies the genericity condition, and f and g are isomorphic via some $T \in \mathrm{GL}(n, q)$, then this T can be decomposed as T_1T_2 for some $T_1 \in W$ from Line 2. Then by the analysis in Step 2, $T_2 = \begin{bmatrix} I_r & 0 \\ 0 & R \end{bmatrix}$ where $R \in \mathrm{Iso}(\mathbf{C}, \mathbf{D})$. When f satisfies the genericity condition, $\mathrm{Iso}(\mathbf{C}, \mathbf{D})$ will be enumerated, so this R will surely be encountered.

To estimate the time complexity of the algorithm, note that $|W| \leq q^{O(rn)}$, and $|\mathrm{Iso}(\mathbf{C}, \mathbf{D})| \leq q^\ell = q^{n-r}$. As other steps are performed in time $\mathrm{poly}(n, m, q)$, enumerating over W and $\mathrm{Iso}(\mathbf{C}, \mathbf{D})$ dominates the time complexity. Recall that $r = 8$. So the total time complexity is upper bounded by $q^{O(n)}$. ◀

3.2 Trilinear form equivalence and algebra isomorphism

We describe out results on trilinear form equivalence and algebra isomorphism, and leave the modifications required to achieve these results in Appendix C.

Trilinear form equivalence. The trilinear form equivalence problem was stated in Section 1.2. In algorithms, a trilinear form f is naturally represented as a 3-way array $\mathtt{A} = (a_{i,j,k})$ where $a_{i,j,k} = f(e_i, e_j, e_k)$. A random trilinear form over $\mathbb{F}_q$ denotes the setting when $\alpha_{i,j,k}$ are independently sampled from $\mathbb{F}_q$ in uniform random.

▶ **Proposition 7.** *Let $f : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ be a random trilinear form, and let $g : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ be an one. There exists a deterministic algorithm that decides whether f and g are equivalent in time $q^{O(n)}$, for all but at most $\frac{1}{q^{\Omega(n)}}$ fraction of f.*

Algebra isomorphism. Let V be a vector space. An algebra is a bilinear map $* : V \times V \to V$. This bilinear map $*$ is considered as the product. Algebras most studied are those with certain conditions on the product, including unital ($\exists v \in V$ such that $\forall u \in V$, $v * u = u$), associative ($(u * v) * w = u * (v * w)$), and commutative ($u * v = v * u$). The authors of [1, 2] study algebras satisfying these conditions. Here we consider algebras without such restrictions. Two algebras $*, \cdot : V \times V \to V$ are *isomorphic*, if there exists $T \in \mathrm{GL}(V)$, such that $\forall u, v \in V$, $T(u) * T(v) = T(u \cdot v)$. As customary in computational algebra, an algebra is represented by its structure constants, i.e. suppose $V \cong \mathbb{F}^n$, and fix a basis $\{e_1, \ldots, e_n\}$. Then $e_i * e_j = \sum_{k\in[n]} \alpha_{i,j,k} e_k$, and this 3-way array $\mathtt{A} = (\alpha_{i,j,k})$ records the structure constants of the algebra with product $*$. A random algebra over $\mathbb{F}_q$ denotes the setting when $\alpha_{i,j,k}$ are independently sampled from $\mathbb{F}_q$ in uniform random.

▶ **Proposition 8.** *Let $f : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q^n$ be a random algebra, and let $g : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q^n$ be an arbitrary algebra. There exists a deterministic algorithm that decides whether f and g are isomorphic in time $q^{O(n)}$, for all but at most $\frac{1}{q^{\Omega(n)}}$ fraction of f.*

4 Complexity of symmetric and alternating trilinear form equivalence

As mentioned in Section 1.4, the proof of Theorem 2 follows by showing that symmetric and alternating trilinear form equivalence are TI-hard (recall Definition 3). In the following we focus on the alternating case. The symmetric case can be tackled in a straightforward way, by starting from the TI-complete problem, symmetric matrix tuple pseudo-isometry, from [11, Theorem B], and modifying the alternating gadget to a symmetric one.

▶ **Proposition 9.** *The alternating trilinear form equivalence problem is* TI*-hard.*

Proof. The starting TI-complete problem. We use the following TI-complete problem from [11]. Let $\mathbf{A} = (A_1, \ldots, A_m), \mathbf{B} = (B_1, \ldots, B_m) \in \Lambda(n, \mathbb{F})^m$ be two tuples of alternating matrices. We say that $\mathbf{A}$ and $\mathbf{B}$ are pseudo-isometric, if there exist $C \in \mathrm{GL}(n, \mathbb{F})$ and $D = (d_{i,j}) \in \mathrm{GL}(m, \mathbb{F})$, such that for any $i \in [m]$, $C^{\mathrm{t}}(\sum_{j\in[m]} d_{i,j} A_j)C = B_i$. By [11, Theorem B], the alternating matrix tuple pseudo-isometry problem is TI-complete. Without loss of generality, we assume that $\dim(\langle A_i \rangle) = \dim(\langle B_i \rangle)$, as if not, then they cannot be pseudo-isometric, and this dimension condition is easily checked.

An alternating trilinear form $\phi : \mathbb{F}^n \times \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ naturally corresponds to a 3-way array $\mathtt{A} = (a_{i,j,k}) \in \mathrm{M}(n \times n \times n, \mathbb{F})$, where $a_{i,j,k} = \phi(e_i, e_j, e_k)$. Then $\mathtt{A}$ is also alternating, i.e. $a_{i,j,k} = 0$ if $i = j$ or $i = k$ or $j = k$, and $a_{i,j,k} = \mathrm{sgn}(\sigma) a_{\sigma(i),\sigma(j),\sigma(k)}$ for any $\sigma \in \mathrm{S}_3$. So in the following, we present a construction of an alternating 3-way array from an alternating matrix tuple, in such a way that two alternating matrix tuples are pseudo-isometric if and only if the corresponding alternating trilinear forms are equivalent.

Constructing alternating 3-way arrays from alternating matrix tuples. Given $\mathbf{A} = (A_1, \ldots, A_m) \in \Lambda(n, \mathbb{F})^m$, construct a 3-way array $\mathtt{A} \in \mathrm{M}(n \times n \times m, \mathbb{F})$, whose ith frontal slice is A_i. Then construct a 3-way array $\mathtt{A}' \in \mathrm{M}(n \times m \times n, \mathbb{F})$, whose ith frontal slice is the ith vertical slice of $\mathtt{A}$. Also construct a 3-way array $\mathtt{A}'' \in \mathrm{M}(m \times n \times n, \mathbb{F})$, whose ith frontal slice is the transpose of the ith horizontal slice of $\mathtt{A}$.

▶ **Example 10** (Running example.)**.** Let us examine a simple example as follows. Let $\mathbf{A} = (A) \in \Lambda(2, \mathbb{F})^1$, where $A = \begin{bmatrix} 0 & a \\ -a & 0 \end{bmatrix}$. Then we have $\mathtt{A} = (A)$; $\mathtt{A}' = (A'_1, A'_2) \in \mathrm{M}(2 \times 1 \times 2, \mathbb{F})$, where $A'_1 = \begin{bmatrix} 0 \\ -a \end{bmatrix}$, and $A'_2 = \begin{bmatrix} a \\ 0 \end{bmatrix}$; $\mathbf{A}'' = (A''_1, A''_2) \in \mathrm{M}(1 \times 2 \times 2, \mathbb{F})$, where $A''_1 = \begin{bmatrix} 0 & a \end{bmatrix}$, and $A''_2 = \begin{bmatrix} -a & 0 \end{bmatrix}$.

From the above $\mathtt{A}$, $\mathtt{A}'$, and $\mathtt{A}''$, we construct $\tilde{\mathtt{A}} \in \mathrm{M}((n+m) \times (n+m) \times (n+m), \mathbb{F})$ as follows. First, we divide $\tilde{\mathtt{A}}$ into the following eight blocks. That is, set $\tilde{\mathtt{A}} = (\tilde{\mathtt{A}}_1, \tilde{\mathtt{A}}_2)$, where $\tilde{\mathtt{A}}_1 = \begin{bmatrix} \tilde{\mathtt{A}}_{1,1,1} & \tilde{\mathtt{A}}_{1,2,1} \\ \tilde{\mathtt{A}}_{2,1,1} & \tilde{\mathtt{A}}_{2,2,1} \end{bmatrix}$, and $\tilde{\mathtt{A}}_2 = \begin{bmatrix} \tilde{\mathtt{A}}_{1,1,2} & \tilde{\mathtt{A}}_{1,2,2} \\ \tilde{\mathtt{A}}_{2,1,2} & \tilde{\mathtt{A}}_{2,2,2} \end{bmatrix}$. Furthermore, $\tilde{\mathtt{A}}_{1,1,1} \in \mathrm{M}(n \times n \times n, \mathbb{F})$, and $\tilde{\mathtt{A}}_{2,2,2} \in \mathrm{M}(m \times m \times m, \mathbb{F})$. Then the sizes of the rest $\tilde{\mathtt{A}}_{i,j,k}$ can be determined accordingly. Now set $\tilde{\mathtt{A}}_{1,1,1}$, $\tilde{\mathtt{A}}_{2,2,1}$, $\tilde{\mathtt{A}}_{1,2,2}$, $\tilde{\mathtt{A}}_{2,1,2}$, and $\tilde{\mathtt{A}}_{2,2,2}$ to be all-zero. Then set $\tilde{\mathtt{A}}_{1,2,1}$ to be $\mathtt{A}'$, $\tilde{\mathtt{A}}_{2,1,1}$ to be $\mathtt{A}''$, and $\tilde{\mathtt{A}}_{1,1,2}$ to be $-\mathtt{A}$. To summarise, we have $\tilde{\mathtt{A}}_1 = \begin{bmatrix} 0 & \mathtt{A}' \\ \mathtt{A}'' & 0 \end{bmatrix}$, and $\tilde{\mathtt{A}}_2 = \begin{bmatrix} -\mathtt{A} & 0 \\ 0 & 0 \end{bmatrix}$.

We claim that $\tilde{\mathtt{A}}$ is alternating. To verify this is straightforward, but somewhat tedious. For example, consider (i, j, k) where $i \in [n]$, $j \in [n+1, n+m]$, and $k \in [n]$. Then $\tilde{\mathtt{A}}(i, j, k) = \mathtt{A}'(i, j-n, k) = \mathtt{A}(i, k, j-n) = -\tilde{\mathtt{A}}(i, k, j)$. We will then need to consider all six permutations of such (i, j, k). In fact, a full proof of all the cases can be extracted from the following example easily.

▶ **Example 11** (Running example, continued from Example 10)**.** We can write out the frontal slices of $\tilde{\mathtt{A}}$ in this case explicitly, which are three alternating matrices $\tilde{A}_1, \tilde{A}_2, \tilde{A}_3 \in \Lambda(3, \mathbb{F})$. That is, $\tilde{A}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -a \\ 0 & a & 0 \end{bmatrix}$, $\tilde{A}_2 = \begin{bmatrix} 0 & 0 & a \\ 0 & 0 & 0 \\ -a & 0 & 0 \end{bmatrix}$, and $\tilde{A}_3 = \begin{bmatrix} 0 & -a & 0 \\ a & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$. It can be verified easily that $\tilde{\mathtt{A}} = (a_{i,j,k})$ is alternating: the nonzero entries are $a_{2,3,1} = -a$, $a_{3,2,1} = a$, $a_{1,3,2} = a$, $a_{3,1,2} = -a$, $a_{1,2,3} = -a$, and $a_{2,1,3} = a$, which are consistent with the signs of the permutations.

The gadget construction. We now describe the gadget construction. The gadget can be described as a block 3-way array as follows. Construct a 3-way array $\mathtt{G}$ of size $(n+1)^2 \times (n+1)^2 \times (n+m)$ over $\mathbb{F}$ as follows. For $i \in [n]$, the ith frontal slice of $\mathtt{G}$ is $\begin{bmatrix} 0 & 0 & \dots & 0 & I_{n+1} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ -I_{n+1} & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{bmatrix}$, where 0 here denotes the $n \times n$ all-zero matrix, I_{n+1} is at the $(1, i+1)$th position, and $-I_{n+1}$ is at the $(i+1, 1)$th position. For $n+1 \le i \le n+m$, the ith frontal slice of $\mathtt{G}$ is the all-zero matrix. We also need the following 3-way arrays derived from $\mathtt{G}$. First, construct $\mathtt{G}'$ of size $(n+m) \times (n+1)^2 \times (n+1)^2$, whose ith horizontal slice is the ith frontal slice of $\mathtt{G}$. Second, construct $\mathtt{G}''$ of size $(n+1)^2 \times (n+m) \times (n+1)^2$, whose ith vertical slice is the ith frontal slice of $\mathtt{G}$.

Now construct a 3-tensor $\hat{\mathtt{A}}$ from $\tilde{\mathtt{A}}$, $\mathtt{G}$, $\mathtt{G}'$, and $\mathtt{G}''$ as follows. The side lengths of $\hat{\mathtt{A}}$ are all equal to $n + m + (n+1)^2$. First, $\hat{\mathtt{A}}$ is divided into eight blocks. That is, $\hat{\mathtt{A}} =$

$(\hat{\mathtt{A}}_1, \hat{\mathtt{A}}_2)$, where $\hat{\mathtt{A}}_1 = \begin{bmatrix} \hat{\mathtt{A}}_{1,1,1} & \hat{\mathtt{A}}_{1,2,1} \\ \hat{\mathtt{A}}_{2,1,1} & \hat{\mathtt{A}}_{2,2,1} \end{bmatrix}$, and $\hat{\mathtt{A}}_2 = \begin{bmatrix} \hat{\mathtt{A}}_{1,1,2} & \hat{\mathtt{A}}_{1,2,2} \\ \hat{\mathtt{A}}_{2,1,2} & \hat{\mathtt{A}}_{2,2,2} \end{bmatrix}$. Furthermore, $\hat{\mathtt{A}}_{1,1,1}$ is of size $(n+m) \times (n+m) \times (n+m)$, $\hat{\mathtt{A}}_{2,2,2}$ is of size $(n+1)^2 \times (n+1)^2 \times (n+1)^2$, and the sizes of the rest $\hat{\mathtt{A}}_{i,j,k}$ can be determined accordingly. Set $\hat{\mathtt{A}}_{1,1,1} = \tilde{\mathtt{A}}$, $\hat{\mathtt{A}}_{1,2,2} = -\mathtt{G}$, $\hat{\mathtt{A}}_{1,2,2} = \mathtt{G}'$, $\hat{\mathtt{A}}_{2,1,2} = \mathtt{G}''$, and the other $\hat{\mathtt{A}}_{i,j,k}$ to be all-zero. To summarise, $\hat{\mathtt{A}}_1 = \begin{bmatrix} \tilde{\mathtt{A}} & 0 \\ 0 & -\mathtt{G} \end{bmatrix}$, and $\hat{\mathtt{A}}_2 = \begin{bmatrix} 0 & \mathtt{G}' \\ \mathtt{G}'' & 0 \end{bmatrix}$.

We claim that $\hat{\mathtt{A}}$ is alternating. To verify this is straightforward but somewhat tedious. So we use the following example from which a complete proof can be extracted easily.

▶ **Example 12.** Consider a 3-tensor $\mathtt{H} = (H)$ be of size $2 \times 2 \times 1$, where $H = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$. Following the recipes of constructing $\mathtt{G}'$ and $\mathtt{G}''$, we construct $\mathtt{H}' = (H'_1, H'_2)$ of size $1 \times 2 \times 2$, where $H'_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}$, and $H'_2 = \begin{bmatrix} -1 & 0 \end{bmatrix}$. And $\mathtt{H}'' = (H''_1, H''_2)$ is of size $2 \times 1 \times 2$, where $H''_1 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$, and $H''_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. Following the recipe of constructing $\hat{\mathtt{A}}$, we construct $\mathtt{C} = (C_1, C_2, C_3)$, where $C_1 = \begin{bmatrix} 0 & 0 \\ 0 & -H \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$, $C_2 = \begin{bmatrix} 0 & H'_1 \\ H''_1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}$, and $C_3 = \begin{bmatrix} 0 & H'_2 \\ H''_2 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$. Then $\mathtt{C} = (c_{i,j,k})$ is alternating, as $c_{2,3,1} = -1$, $c_{3,2,1} = 1$, $c_{1,3,2} = 1$, $c_{3,1,2} = -1$, $c_{1,2,3} = -1$, $c_{2,1,3} = 1$, which are consistent with the signs of the permutations.

Proof of correctness. Let $\mathbf{A}, \mathbf{B} \in \Lambda(n, \mathbb{F})^m$. Let $\hat{\mathtt{A}} = (\begin{bmatrix} \tilde{\mathtt{A}} & 0 \\ 0 & -\mathtt{G} \end{bmatrix}, \begin{bmatrix} 0 & \mathtt{G}' \\ \mathtt{G}'' & 0 \end{bmatrix})$, $\hat{\mathtt{B}} = (\begin{bmatrix} \tilde{\mathtt{B}} & 0 \\ 0 & -\mathtt{G} \end{bmatrix}, \begin{bmatrix} 0 & \mathtt{G}' \\ \mathtt{G}'' & 0 \end{bmatrix}) \in \mathrm{M}((n+m+(n+1)^2) \times (n+m+(n+1)^2) \times (n+m+(n+1)^2), \mathbb{F})$ be constructed from $\mathbf{A}$ and $\mathbf{B}$ using the procedure above, respectively.

We claim that $\mathbf{A}$ and $\mathbf{B}$ are pseudo-isometric if and only if $\hat{\mathtt{A}}$ and $\hat{\mathtt{B}}$ are equivalent as trilinear forms.

The only if direction. Suppose $P^{\mathrm{t}}\mathbf{A}P = \mathbf{B}^Q$ for some $P \in \mathrm{GL}(n, \mathbb{F})$ and $Q \in \mathrm{GL}(m, \mathbb{F})$.

We will construct a trilinear form equivalence from $\hat{\mathtt{A}}$ to $\hat{\mathtt{B}}$ of the form $S = \begin{bmatrix} P & 0 & 0 \\ 0 & Q^{-1} & 0 \\ 0 & 0 & R \end{bmatrix} \in \mathrm{GL}(n+m+(n+1)^2, \mathbb{F})$, where $R \in \mathrm{GL}((n+1)^2, \mathbb{F})$ is to be determined later on.

Recall that $\hat{\mathtt{A}} = (\begin{bmatrix} \tilde{\mathtt{A}} & 0 \\ 0 & -\mathtt{G} \end{bmatrix}, \begin{bmatrix} 0 & \mathtt{G}' \\ \mathtt{G}'' & 0 \end{bmatrix})$, $\hat{\mathtt{B}} = (\begin{bmatrix} \tilde{\mathtt{B}} & 0 \\ 0 & -\mathtt{G} \end{bmatrix}, \begin{bmatrix} 0 & \mathtt{G}' \\ \mathtt{G}'' & 0 \end{bmatrix})$. It can be verified that the action of S sends $\tilde{\mathtt{A}}$ to $\tilde{\mathtt{B}}$. It remains to show that, by choosing an appropriate R, the action of S also sends $\mathtt{G}$ to $\mathtt{G}$.

Let $\mathtt{G}_1$ be the first n frontal slices of $\mathtt{G}$, and $\mathtt{G}_2$ the last m frontal slices from $\mathtt{G}$. Then the action of S sends $\mathtt{G}_1$ to $R^{\mathrm{t}}\mathtt{G}_1^P R$, and $\mathtt{G}_2$ to $R^{\mathrm{t}}\mathtt{G}_2^{Q^{-1}} R$. Since $\mathtt{G}_2$ is all-zero, the action of S on $\mathtt{G}_2$ results in an all-zero tensor, so we have $R^{\mathrm{t}}\mathtt{G}_2^{Q^{-1}} R = \mathtt{G}_2$.

We then turn to $\mathtt{G}_1$. For $i \in [n+1]$, consider the ith horizontal slice of $\mathtt{G}_1$, which is of the form $H_i = \begin{bmatrix} 0 & B_{1,i} & B_{2,i} & \dots & B_{n,i} \end{bmatrix}$, where O denotes the $n \times (n+1)$ all-zero matrix, and $B_{j,i}$ is the $n \times (n+1)$ elementary matrix with the (j,i)th entry being 1, and other entries being 0. Note that those non-zero entries of H_i are in the $(k(n+1)+i)$th columns, for $k \in [n]$. Let $P^{\mathrm{t}} = \begin{bmatrix} p_1 & \dots & p_n \end{bmatrix}$, where p_i is the ith column of P^{t}. Then P acts on H_i from

the left, which yields $P^{\mathrm{t}}H_i = \begin{bmatrix} 0 & P_{1,i} & \dots & P_{n,i} \end{bmatrix}$, where $P_{j,i}$ denotes the $n \times (n+1)$ matrix with the ith column being p_j, and the other columns being 0.

Let us first set $R = \begin{bmatrix} I_{n+1} & 0 \\ 0 & \hat{R} \end{bmatrix}$, where $\hat{R}$ is to be determined later on. Then the left action of R on $\mathtt{G}_1$ preserves H_i through I_{n+1}. The right action of R on $\mathtt{G}_1$ translates to the right action of $\hat{R}$ on H_i. To send $P^{\mathrm{t}}H_i$ back to H_i, $\hat{R}$ needs to act on those $(k(n+1)+i)$th columns of H_i, $i \in [n+1]$, as P^{-1}. Note that for H_i and H_j, $i \neq j$, those columns with non-zero entries are disjoint. This gives $\hat{R}$ the freedom to handle different H_i's separately. In other words, $\hat{R}$ can be set as $P^{-1} \otimes I_{n+1}$. This ensures that for every H_i, $P^{\mathrm{t}}H_i\hat{R} = H_i$. To summarise, we have $R^{\mathrm{t}}\mathtt{G}_1^P R = \mathtt{G}_1$, and this concludes the proof for the only if direction.

The if direction. Suppose $\hat{\mathtt{A}}$ and $\hat{\mathtt{B}}$ are isomorphic as trilinear forms via $P \in \mathrm{GL}(n+m+(n+1)^2, \mathbb{F})$. Set $P = \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix}$, where $P_{1,1}$ is of size $n \times n$, $P_{2,2}$ is of size $m \times m$, and $P_{3,3}$ is of size $(n+1)^2 \times (n+1)^2$. Consider the ranks of the frontal slices of $\hat{\mathtt{A}}$.

- The ranks of the first n frontal slices are in $[2(n+1), 4n]$. This is because a frontal slice in this range consists of two copies of vertical slices of $\mathtt{A}$ (whose ranks are between $[0, n-1]$ due to the alternating condition), and one frontal slice of $\mathtt{G}$ (whose ranks are of $2(n+1)$).
- The ranks of the $n+1$ to $n+m$ frontal slices are in $[0, n]$. This is because a frontal slice in this range consists of only just one frontal slice of $\mathtt{A}$.
- The ranks of the last $n(n+1)$ vertical slices are in $[0, 2n]$. This is because a frontal slice in this range consists of two copies of horizontal slices of $\mathtt{G}$ (whose ranks are either n or 1; see e.g. the form of H_i in the proof of the only if direction).

By the discussions above, we claim that that P must be of the form $\begin{bmatrix} P_{1,1} & 0 & 0 \\ P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix}$. To see this, for the sake of contradiction, suppose there are non-zero entries in $P_{1,2}$ or $P_{1,3}$. Then a non-trivial linear combination of the first n frontal slices is added to one of the last $(m+(n+1)^2)$ frontal slices. This implies that for this slice, the lower-right $(n+1)^2 \times (n+1)^2$ submatrix is of the form $\begin{bmatrix} 0 & a_1 I_{n+1} & a_2 I_{n+1} & \dots & a_n I_{n+1} \\ -a_1 I_{n+1} & 0 & 0 & \dots & 0 \\ -a_2 I_{n+1} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_n I_{n+1} & 0 & 0 & \dots & 0 \end{bmatrix}$, where one of $a_i \in \mathbb{F}$ is non-zero. Then this slice is of rank $\geq 2(n+1)$, which is unchanged by left (resp. right) multiplying P^{t} (resp. P), so it cannot be equal to the corresponding slice of $\hat{\mathtt{B}}$ which is of rank $\leq 2n$. We then arrived at the desired contradiction.

Now consider the action of such P on the $n+1$ to $n+m$ frontal slices. Note that these slices are of the form $\begin{bmatrix} A_i & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$. (Recall that the last m slices of $\mathtt{G}$ are all-zero matrices.) Then we have $\begin{bmatrix} P_{1,1}^{\mathrm{t}} & P_{2,1}^{\mathrm{t}} & P_{3,1}^{\mathrm{t}} \\ 0 & P_{2,2}^{\mathrm{t}} & P_{3,2}^{\mathrm{t}} \\ 0 & P_{2,3}^{\mathrm{t}} & P_{3,3}^{\mathrm{t}} \end{bmatrix} \begin{bmatrix} A_i & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_{1,1} & 0 & 0 \\ P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix} = \begin{bmatrix} P_{1,1}^{\mathrm{t}} A_i P_{1,1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$. Since $P^{\mathrm{t}}\hat{\mathtt{A}}^P P = \hat{\mathtt{B}}$, we have $P^{\mathrm{t}}\hat{\mathtt{A}}P = \hat{\mathtt{B}}^{P^{-1}}$. Observe that for the upper-left $n \times n$ submatrices of the frontal slices of $\hat{\mathtt{B}}$, P^{-1} simply performs a linear combination of B_i's. It follows that every $P_{1,1}^{\mathrm{t}} A_i P_{1,1}$ is in the linear span of B_i. Since we assumed $\dim(\langle A_i \rangle) = \dim(\langle B_i \rangle)$, we have that $\mathbf{A}$ and $\mathbf{B}$ are pseudo-isometric. This concludes the proof of Proposition 9. ◀

References

1. Manindra Agrawal and Nitin Saxena. Automorphisms of finite rings and applications to complexity of problems. In *STACS 2005, 22nd Annual Symposium on Theoretical Aspects of Computer Science, Proceedings*, pages 1–17, 2005.
2. Manindra Agrawal and Nitin Saxena. Equivalence of $\mathbb{F}$-algebras and cubic forms. In *STACS 2006, 23rd Annual Symposium on Theoretical Aspects of Computer Science, Proceedings*, pages 115–126, 2006.
3. Jérémy Berthomieu, Jean-Charles Faugère, and Ludovic Perret. Polynomial-time algorithms for quadratic isomorphism of polynomials: The regular case. *J. Complexity*, 31(4):590–616, 2015.
4. Charles Bouillaguet. *Etudes d'hypothèses algorithmiques et attaques de primitives cryptographiques*. PhD thesis, PhD thesis, Université Paris-Diderot–École Normale Supérieure, 2011.
5. Charles Bouillaguet, Jean-Charles Faugère, Pierre-Alain Fouque, and Ludovic Perret. Practical cryptanalysis of the identification scheme based on the isomorphism of polynomial with one secret problem. In *International Workshop on Public Key Cryptography*, pages 473–493. Springer, 2011.
6. Peter A. Brooksbank, Yinan Li, Youming Qiao, and James B. Wilson. Improved algorithms for alternating matrix space isometry: From theory to practice. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 26:1–26:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
7. Peter A. Brooksbank and James B. Wilson. Computing isometry groups of Hermitian maps. *Trans. Amer. Math. Soc.*, 364:1975–1996, 2012.
8. Vyacheslav Futorny, Joshua A. Grochow, and Vladimir V. Sergeichuk. Wildness for tensors. *Lin. Alg. Appl.*, 566:212–244, 2019.
9. Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.
10. Joshua A. Grochow and Youming Qiao. Algorithms for group isomorphism via group extensions and cohomology. *SIAM J. Comput.*, 46(4):1153–1216, 2017. Preliminary version in IEEE Conference on Computational Complexity (CCC) 2014 (DOI:10.1109/CCC.2014.19). Also available as arXiv:1309.1776 [cs.DS] and ECCC Technical Report TR13-123.
11. Joshua A. Grochow and Youming Qiao. Isomorphism problems for tensors, groups, and cubic forms: completeness and reductions. *CoRR*, abs/1907.00309, 2019.
12. Gábor Ivanyos and Youming Qiao. Algorithms based on *-algebras, and their applications to isomorphism of polynomials with one secret, group isomorphism, and polynomial identity testing. *SIAM J. Comput.*, 48(3):926–963, 2019.
13. Neeraj Kayal. Efficient algorithms for some special cases of the polynomial equivalence problem. In *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, 2011.
14. Serge Lang. *Algebra*. Number 211 in Graduate Texts in Mathematics. Springer-Verlag, New York, third enlarged edition, 2002.
15. Yinan Li and Youming Qiao. Linear algebraic analogues of the graph isomorphism problem and the Erdős–Rényi model. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 463–474. IEEE Computer Society, 2017. arXiv:1708.04501, version 2.
16. Eugene M. Luks. Permutation groups and polynomial-time computation. In *Groups and computation (New Brunswick, NJ, 1991)*, volume 11 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 139–175. Amer. Math. Soc., Providence, RI, 1993.
17. Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In *Advances in Cryptology - EUROCRYPT*

'96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding, pages 33–48, 1996.

18 Nitin Saxena. *Morphisms of rings and applications to complexity.* PhD thesis, Indian Institute of Technology, Kanpur, May 2006.

19 H. Weyl. *The classical groups: their invariants and representations*, volume 1. Princeton University Press, 1997.

20 R. Wilson. *The Finite Simple Groups*, volume 251 of *Graduate Texts in Mathematics*. Springer London, 2009.

A Proof of Lemma 6

Recall that $\mathbf{C} = (C_1, \ldots, C_8) \in \mathrm{S}(\ell, q)^8$ is a tuple of random symmetric matrices, and $\mathrm{Aut}(\mathbf{C}) = \{R \in \mathrm{GL}(\ell, q) : R^{\mathrm{t}}\mathbf{C}R = \mathbf{C}\}$. Our goal is to prove that $|\mathrm{Aut}(\mathbf{C})| \leq q^{\ell}$ for all but at most $\frac{1}{q^{\Omega(\ell)}}$ fraction of random $\mathbf{C}$.

Let $\mathrm{Adj}(\mathbf{C}) := \{(R, S) \in \mathrm{M}(\ell, q) \oplus \mathrm{M}(\ell, q) : R^{\mathrm{t}}\mathbf{C} = \mathbf{C}S\}$. It is clear that $|\mathrm{Aut}(\mathbf{C})| \leq |\mathrm{Adj}(\mathbf{C})|$. We will in fact prove that with high probability, $|\mathrm{Adj}(\mathbf{C})| \leq q^{\ell}$. The proof of the following mostly follows the proofs for general matrix spaces as in [15] and alternating matrix spaces as in [6].

To start with, we make use the following result from [15]. We say that $\mathbf{D} = (D_1, \ldots, D_r) \in \mathrm{M}(\ell, q)^r$ is *stable*, if for any $U \leq \mathbb{F}_q^{\ell}$, $1 \leq \dim(U) \leq \ell - 1$, $\dim(\mathbf{D}(U)) > \dim(U)$, where $\mathbf{D}(U) = \langle \cup_{i \in [r]} D_i(U) \rangle$.

▷ Claim 13 ([15, Prop. 17]). If $\mathbf{D} \leq \mathrm{M}(\ell, q)$ is stable, then $|\mathrm{Adj}(\mathbf{D})| \leq q^{\ell}$.

Therefore, we turn to show that a random $\mathbf{C} \in \mathrm{S}(\ell, q)^8$ is stable with high probability. This was shown for random matrix tuples in $\mathrm{M}(\ell, q)^4$ in [15], and random alternating matrix tuples in $\Lambda(\ell, q)^{16}$ in [6]. The proof strategy for the symmetric case is similar, but certain differences between the symmetric and alternating matrices do arise, as reflected in the following.

Our goal is to show that

$$\Pr[\mathbf{C} \in \mathrm{S}(\ell, q)^8 \text{ is not stable}] \leq \frac{1}{q^{\Omega(n)}}.$$

By definition, we have

$$\Pr[\mathbf{C} \in \mathrm{S}(\ell, q)^8 \text{ is not stable}] = \Pr[\exists U \leq \mathbb{F}_q^{\ell}, 1 \leq \dim(U) \leq n-1, \dim(U) \geq \dim(\mathbf{C}(U))].$$

By union bound, we have

$$\begin{aligned} &\Pr[\exists U \leq \mathbb{F}_q^{\ell}, 1 \leq \dim(U) \leq n-1, \dim(U) \geq \dim(\mathbf{C}(U))] \\ \leq &\sum_{U \leq \mathbb{F}_q^{\ell}, 1 \leq \dim(U) \leq n-1} \Pr[\dim(U) \geq \dim(\mathbf{C}(U))]. \end{aligned}$$

For $d \in [n-1]$, let $E_d = \langle e_1, \ldots, e_d \rangle$. Let $U \leq \mathbb{F}_q^{\ell}$, $\dim(U) = d$. We claim that $\Pr[\dim(U) \geq \dim(\mathbf{C}(U))] = \Pr[\dim(E_d) \geq \dim(\mathbf{C}(E_d))]$. To see this, note that there exists $P \in \mathrm{GL}(\ell, q)$ such that $P(E_d) = U$. Then observe that $\dim((P^{\mathrm{t}}\mathbf{C}P)(E_d)) = \dim(\mathbf{C}(U))$. It follows that $\dim(\mathbf{C}(U)) \leq \dim(U)$ if and only if $\dim((P^{\mathrm{t}}\mathbf{C}P)(E_d)) \leq \dim(E_d)$. The claim then follows, by observing that the map $\mathrm{S}(\ell, q)^r \to \mathrm{S}(\ell, q)^r$ via $P^{\mathrm{t}} \cdot P$ is bijective. As a consequence, for any $d \in [n-1]$, we have

$$\sum_{U \leq \mathbb{F}_q^{\ell}, \dim(U) = d} \Pr[\dim(U) \geq \dim(\mathbf{C}(U))] = \begin{bmatrix} \ell \\ d \end{bmatrix}_q \cdot \Pr[\dim(\mathbf{C}(E_d)) \leq d].$$

Let C_i^d be the submatrix of C_i consisting of the first d columns of C_i, and let $C^d = \begin{bmatrix} C_1^d & \cdots & C_r^d \end{bmatrix} \in \mathrm{M}(\ell \times rd, q)$. Then $\dim(\mathbf{C}(E_d)) = \mathrm{rk}(C^d)$. Note that each C_i^d is of the form

$$\begin{bmatrix} C_{i,1}^d \\ C_{i,2}^d \end{bmatrix} \tag{5}$$

where $C_{i,1}^d$ is a random symmetric matrix of size $d \times d$, and $C_{i,2}^d$ is a random matrix of size $(n-d) \times d$.

We then need to prove the following result, from which our desired result would follow. Here we set $r = 8$.

▶ **Proposition 14.** *Let $C^d \in \mathrm{M}(\ell \times 8d, q)$ be in the form above. Then we have $\begin{bmatrix} n \\ d \end{bmatrix}_q \cdot \Pr[\mathrm{rk}(C^d)] \leq \frac{1}{q^{\Omega(\ell)}}$,*

To prove Proposition 14, We wish to utilise the following result from [15].

▶ **Proposition 15** ([15, Proposition 20]). *Let $D \in \mathrm{M}(\ell \times 4d, q)$ be a random matrix, where $1 \leq d \leq n-2$. Then $\begin{bmatrix} \ell \\ d \end{bmatrix}_q \cdot \Pr[\mathrm{rk}(D) \leq d] \leq \frac{1}{q^{\Omega(\ell)}}$.*

To use the above result in our setting, however, there is a caveat caused by the symmetric structure of $C_{i,1}^d$ for $i \in [r]$. This is resolved by observing the following claim, which basically says that we can simulate one random matrix in $\mathrm{M}(d, q)$ using two random symmetric matrices in $\mathrm{S}(d, q)$.

▷ Claim 16. Let X and Y be two random symmetric matrices from $\mathrm{S}(d, q)$, i.e.

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{1,2} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1,d} & x_{2,d} & \cdots & x_{d,d} \end{bmatrix}, Y = \begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,d} \\ y_{1,2} & y_{2,2} & \cdots & y_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1,d} & y_{2,d} & \cdots & y_{d,d} \end{bmatrix}$$

Then

$$Z = \begin{bmatrix} x_{1,1} + y_{1,2} & x_{1,2} + y_{1,3} & \cdots & x_{1,d} + y_{1,1} \\ x_{1,2} + y_{2,2} & x_{2,2} + y_{2,3} & \cdots & x_{2,d} + y_{1,2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1,d} + y_{2,d} & x_{2,d} + y_{3,d} & \cdots & x_{d,d} + y_{1,d} \end{bmatrix}$$

is a uniformly sampled random matrix in $\mathrm{M}(d, q)$, when X and Y are sampled in uniformly random from $\mathrm{S}(d, q)$.

Proof. Let $z_{i,j}$ be the (i,j)th entry of Z. Note that each $x_{i,j}$ (resp. $y_{i,j}$), $i \neq j$, appear exactly twice in Z on an antidiagonal $z_{1,i}, z_{2,i-1}, \ldots, z_{i-1,1}, z_{i,n}, z_{i+1,n-1}, \ldots, z_{n,i+1}$. So we can focus on such an antidiagonal to show that when $x_{i,j}$ and $y_{i,j}$ are uniformly sampled from $\mathbb{F}_q$, $z_{i,j}$ are also uniformly sampled from $\mathbb{F}_q$.

Let us first consider the case when d is odd. Let us consider a specific one, say $z_{1,1} = x_{1,1} + y_{1,2}$, $z_{2,d} = x_{2,d} + y_{1,2}$, $\ldots$, $z_{d,2} = x_{2,d} + y_{3,d}$. Other antidiagonals are of the same structure. It can be verified that this is a system of d linear equations in $d+1$ variables of rank d. It follows that when those $x_{i,j}$ and $y_{k,\ell}$ involved are sampled in uniform random from $\mathbb{F}_q$, $z_{i',j'}$ are also in uniformly random distribution.

The case when d is even can be verified similarly. This concludes the proof. ◀

We are now ready to prove Proposition 14.

Proof of Proposition 14. Recall that $C^d = \begin{bmatrix} C_1^d & \dots & C_8^d \end{bmatrix}$, where $C_i^d \in \mathrm{M}(\ell \times d, q)$ is of the form in Equation 5. For $i \in [4]$, let $C_i'^d \in \mathrm{M}(\ell \times d, q)$ be constructed from C_{2i-1}^d, C_{2i}^d as in Claim 16, and set $C'^d = \begin{bmatrix} C_1'^d & \dots & C_4'^d \end{bmatrix}$. It is clear that $\mathrm{rk}(C^d) \geq \mathrm{rk}(C'^d)$, so $\begin{bmatrix} \ell \\ d \end{bmatrix}_q \cdot \Pr[\mathrm{rk}(C^d) \leq d] \leq \begin{bmatrix} \ell \\ d \end{bmatrix}_q \cdot \Pr[\mathrm{rk}(C'^d) \leq d]$. By Claim 16, C'^d is a random matrix in $\mathrm{M}(\ell \times 4d, q)$. By Proposition 15, $\begin{bmatrix} \ell \\ d \end{bmatrix}_q \cdot \Pr[\mathrm{rk}(C'^d) \leq d] \leq \frac{1}{q^{\Omega(\ell)}}$. This concludes the proof. ◀

B Proof of the remaining cases of Theorem 1

Given Proposition 5, we can complete the proof of Theorem 1 easily.

Proof. **Cubic forms over fields of characteristic 2.** In Proposition 5 we solved the case for cubic forms over fields of odd orders. We now consider cubic forms over fields of characteristic 2.

In this case, one difficulty is that the correspondences between quadratic forms and symmetric matrices as used in Equation 4. Still, this difficulty can be overcome as follows. Let $f = \sum_{1 \leq i \leq j \leq k \leq n} \alpha_{i,j,k} x_i x_j x_k$ where $\alpha_{i,j,k} \in \mathbb{F}_q$, q is a power of 2. We follow the proof strategy of Proposition 5. Step 1 stays exactly the same. In Step 2, we have f and g_1, and the question is to look for $T_2 = \begin{bmatrix} I_r & 0 \\ 0 & R \end{bmatrix}$ such that $f = g_1 \circ T_2$. We still consider the quadratic forms $c_i = \sum_{1 \leq j \leq k \leq \ell} \alpha_{i,j,k} y_j y_k$ for $i \in [r]$. Now note that $(\sum_{j \in [\ell]} \beta_j y_j)^2 = \sum_{j \in [\ell]} \beta_j^2 y_j^2$ over fields of characteristic 2. So the monomials y_j^2 do not contribute to $y_j y_k$ for $j \neq k$ under linear transformations. It follows that we can restrict our attention to $c_i' = \sum_{1 \leq j < k \leq \ell} \alpha_{i,j,k}' y_j y_k$ for $i \in [r]$, and define alternating matrices

$$C_i = \begin{bmatrix} 0 & \alpha_{i,1,2}' & \dots & \alpha_{i,1,\ell}' \\ \alpha_{i,1,2}' & 0 & \dots & \alpha_{i,2,\ell}' \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{i,1,\ell}' & \alpha_{i,2,\ell}' & \dots & 0 \end{bmatrix} \tag{6}$$

for $i \in [r]$ to get $\mathbf{C} \in \Lambda(\ell, q)^r$. Note that C_i is alternating because we work over fields of characteristic 2. Similarly construct $\mathbf{D} \in \Lambda(\ell, q)^r$ from g_1. It can then be verified that, for $T_2 = \begin{bmatrix} I_r & 0 \\ 0 & R \end{bmatrix}$ to be an isomorphism from g_1 to f, it is necessary that R is an isometry from $\mathbf{D}$ to $\mathbf{C}$. We then use [6, Proposition 12], which is the alternating matrix version of our Lemma 6. That proposition ensures that for $r = 20$, all but at most $\frac{1}{q^{\Omega(\ell)}}$ fraction of $\mathbf{C}$ has $|\mathrm{Aut}(\mathbf{C})| \leq q^\ell$. This explains how the first difficulty is overcome.

However, there is a second difficulty, namely Theorem 4 do not apply to fields of characteristic 2. We sketch how to overcome this difficulty here. The key is to look into the proof of [6, Proposition 12], which in fact ensures that $\mathrm{Adj}(\mathbf{C}) = \{(A, E) \in \mathrm{M}(\ell, q) \oplus \mathrm{M}(\ell, q) \mid A^{\mathsf{t}}\mathbf{C} = \mathbf{C}E\}$ is of size $\leq q^\ell$ for random $\mathbf{C}$. Note that $\mathrm{Adj}(\mathbf{C})$ is a linear space and a linear basis of $\mathrm{Adj}(\mathbf{C})$ can be solved efficiently. Therefore, replacing $\mathrm{Aut}(\mathbf{C})$ with $\mathrm{Adj}(\mathbf{C})$ and $\mathrm{Iso}(\mathbf{C}, \mathbf{D})$ with $\mathrm{Adj}(\mathbf{C}, \mathbf{D}) = \{(A, E) \in \mathrm{M}(\ell, q) \oplus \mathrm{M}(\ell, q) \mid A^{\mathsf{t}}\mathbf{C} = \mathbf{D}E\}$, we can proceed as in the proof of Proposition 5. The interested readers may refer to [6] for the details.

Degree-d forms. Let us then consider degree-d forms. In this case, we follow the proof of Proposition 5. Step 1 stays exactly the same. In Step 2, instead of $\sum_{1 \leq j \leq k \leq n} \alpha_{i,j,k} x_i x_j x_k$ for $i \in [r]$, we work with $\sum_{1 \leq j \leq k \leq n} \alpha_{i,j,k} x_i^{d-2} x_j x_k$, noting that matrices in the form in Equation 2 preserve the set of monomials $\{x_i^{d-2} x_j x_k\}$. Then for odd q case, construct

symmetric matrices as in Equation 4 and proceed as in the rest of Proposition 5. For the even q case, construct alternating matrices as in Equation 6, and procees as described above.

Degree-d polynomials. We now consider degree-d polynomials. In this case, we can single out the degree-d piece and work as in degree-d form case. The only change is that in the verification step, we need to take into account the monomials of degree $< d$ as well.

This concludes the proof of Theorem 1. ◀

C On Propositions 7 and 8

To test equivalence of trilinear forms of $f, g : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$, an average-case algorithm in time $q^{O(n)}$ can be achieved by following the proof of Proposition 5. The only difference is that, in Step 2 there, instead of symmetric matrices in Equation 4, we can construct general matrices $C_i = (\alpha'_{i,j,k})_{j,k\in[\ell]}$. Then we need a version of Lemma 6 for general matrices, which is already shown in [15, Proposition 19 and 20]. It says that when $r = 4$, a random $\mathbf{C} \in \mathrm{M}(\ell, q)^4$ satisfies that $|\mathrm{Aut}(\mathbf{C})| \leq q^\ell$. We then proceed exactly as in Proposition 5 for odd q, and for even q we use the technique described in Section B.

Suppose we have two algebras $*, \cdot : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q^n$, represented by their structure constants. The proof strategy of Proposition 7 carries out to test algebra isomorphism in a straightforward fashion. The only difference is in the verification step (i.e. Line 3f). More specifically, we can write an algebra as an element in $(\mathbb{F}_q^n)^* \otimes \mathbb{F}_q^n \otimes \mathbb{F}_q^n$, where $(\mathbb{F}_q^n)^*$ is the dual space of $\mathbb{F}_q^n$.[3] It follows that we can write $*$ as $\sum_{i,j,k\in[n]} \alpha_{i,j,k} e_i^* \otimes e_j \otimes e_k$. The key difference with trilinear form equivalence is that for AI, $T \in \mathrm{GL}(n, q)$ acts on e_i^* by its inverse. So the algorithm for AI is the same as the one for trilinear form equivalence, except that in the verification step we need to use R^{-1} instead of R to act on the first argument.

[3] Note that here we put $(\mathbb{F}_q^n)^*$ as the first argument, instead of the last one, in order to be consistent with the procedure in Proposition 7. This is without loss of generality due to the standard isomorphism between $U \otimes V \otimes W$ and $W \otimes U \otimes V$.