Improved Algorithms for Alternating Matrix Space Isometry: From Theory to Practice

Peter A. Brooksbank
Department of Mathematics, Bucknell University, Lewisburg, PA, USA
pbrooksb@bucknell.edu

Yinan Li
CWI and QuSoft, Amsterdam, The Netherlands
liyinan9252@gmail.com

Youming Qiao
Centre for Quantum Software and Information, University of Technology Sydney, Ultimo, Australia
Youming.Qiao@uts.edu.au

James B. Wilson
Department of Mathematics, Colorado State University, Fort Collins, CO, USA
James.Wilson@ColoState.Edu

Abstract

Motivated by testing isomorphism of p-groups, we study the alternating matrix space isometry problem (AltMatSpIso), which asks to decide whether two m-dimensional subspaces of $n \times n$ alternating (skew-symmetric if the field is not of characteristic 2) matrices are the same up to a change of basis. Over a finite field $\mathbb{F}_p$ with some prime $p \neq 2$, solving AltMatSpIso in time $p^{O(n+m)}$ is equivalent to testing isomorphism of p-groups of class 2 and exponent p in time polynomial in the group order. The latter problem has long been considered a bottleneck case for the group isomorphism problem.

Recently, Li and Qiao presented an average-case algorithm for AltMatSpIso in time $p^{O(n)}$ when n and m are linearly related (FOCS '17). In this paper, we present an average-case algorithm for AltMatSpIso in time $p^{O(n+m)}$. Besides removing the restriction on the relation between n and m, our algorithm is considerably simpler, and the average-case analysis is stronger. We then implement our algorithm, with suitable modifications, in Magma. Our experiments indicate that it improves significantly over default (brute-force) algorithms for this problem.

2012 ACM Subject Classification Computing methodologies → Algebraic algorithms; Computing methodologies → Combinatorial algorithms; Mathematics of computing → Graph algorithms

Keywords and phrases Alternating Matrix Spaces, Average-case Algorithm, p-groups of Class 2and Exponent p, Magma

Digital Object Identifier 10.4230/LIPIcs.ESA.2020.26

Related Version The main results of this submission come from [5] (`https://arxiv.org/abs/1905.02518`), which will be subdivided into several papers.

Funding The authors would like to acknowledge the financial support by NSF grant DMS-1750319. Additionally,
Peter A. Brooksbank: Partially supported by NSF grant DMS-1620362.
Yinan Li: Partially supported by ERC Consolidator Grant 615307-QPROGRESS.
Youming Qiao: Partially supported by the Australian Research Council DE150100720 and DP200100950.
James B. Wilson: Partially supported by NSF grant DMS-1620454.

Acknowledgements The authors would like to acknowledge László Babai and Xiaorui Sun for discussions, and Joshua A. Grochow for discussions and comments on improving the presentation. The authors acknowledge the Hausdorff Institute for Mathematics, the University of Auckland, the Santa Fe Institute, and the TACA workshop at the University of Colorado, Boulder and Colorado State University, where this research was carried out by (subsets of) the authors.

© Peter A. Brooksbank, Yinan Li, Youming Qiao, and James B. Wilson;
licensed under Creative Commons License CC-BY

28th Annual European Symposium on Algorithms (ESA 2020).
Editors: Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders; Article No. 26; pp. 26:1–26:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Motivated by testing isomorphism of p-groups, we study the Alternating Matrix Space Isometry (AltMatSpIso) problem. To state this problem, we set up some notation and definitions. Let $[n] = \{1, \dots, n\}$, and let $\mathrm{M}(n, q)$ be the linear space of $n \times n$ matrices over the finite field $\mathbb{F}_q$ (q is a prime power). A matrix $A \in \mathrm{M}(n, q)$ is *alternating* if for any vector $v \in \mathbb{F}_q^n$, $v^{\mathrm{t}}Av = 0$, where v^{t} denote the transpose of v. (When q is odd, A is alternating if and only if A is *skew-symmetric*, i.e. $A^{\mathrm{t}} = -A$.) Let $\Lambda(n, q)$ be the linear space of all alternating matrices in $\mathrm{M}(n, \mathbb{F}_q)$. Let $\mathcal{A}, \mathcal{B}$ be subspaces of $\Lambda(n, q)$. We say that $\mathcal{A}$ and $\mathcal{B}$ are *isometric*, if there exists an invertible matrix $T \in \mathrm{GL}(n, q)$, such that $T^{\mathrm{t}}\mathcal{A}T := \{T^{\mathrm{t}}AT : A \in \mathcal{A}\}$ and $\mathcal{B}$ are the same subspace. The AltMatSpIso problem then asks, given linear bases of two alternating matrix spaces $\mathcal{A}$ and $\mathcal{B}$, decide whether $\mathcal{A}$ and $\mathcal{B}$ are isometric.

As we will elaborate in detail in Subsection 1.1, the AltMatSpIso problem has been studied for decades. Indeed, it lies at the heart of the Group Isomorphism (GpI) problem, and has an intimate relationship with many other isomorphism problems, including Graph Isomorphism (GI). As a problem in $\mathsf{NP} \cap \mathsf{coAM}$, its worst-case time complexity has barely been improved over the brute-force algorithm, which enumerates all invertible matrices in $\mathrm{GL}(n, q)$ to search for some $T \in \mathrm{GL}(n, q)$ satisfying $T^{\mathrm{t}}\mathcal{A}T = \mathcal{B}$. The brute-force algorithm takes time $q^{\Theta(n^2)} \cdot \mathrm{poly}(n, m, \log q)$ in the worst case. To obtain an algorithm in time $q^{O(n+m)}$ would lead to an algorithm testing isomorphism of *p-groups of class 2 and exponent p* in time polynomial *in the group order*, which is a long-standing open problem.

Recently, Li and Qiao developed an *average-case* algorithm to tackle the AltMatSpIso problem, *when m and n are linearly related* [21]. The algorithm takes time $q^{O(n)}$ and tests isometry between all but at most $q^{-\Omega(n)}$ fraction of $\mathcal{A}$ in $\Lambda(n, q)$, and an arbitrary $\mathcal{B}$. Here a random m-dimensional $n \times n$ alternating matrix space $\mathcal{A}$ is chosen with probability $\begin{bmatrix} \binom{n}{2} \\ m \end{bmatrix}_q^{-1}$, where $\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}_q$ denotes the q-Gaussian binomial coefficient and $\begin{bmatrix} \binom{n}{2} \\ m \end{bmatrix}_q$ is the total number of m-dimensional alternating matrix spaces in $\Lambda(n, q)$. This may be viewed as a linear algebraic analogue of the *Erdős-Rényi model* for graphs [10]. A key idea behind their algorithm is to view the AltMatSpIso problem as a linear algebraic analogue of the GI problem [21], which leads to adapting the individualisation and refinement technique for random graph isomorphism as used by Babai, Erdős, and Selkow in [2]. We summarise these algorithms in Subsection 1.2. For convenience, we shall refer to the algorithm from [21] as the Li-Qiao algorithm.

In this paper, we present another average-case algorithm for AltMatSpIso.

▶ **Theorem 1.** *Suppose $m \geqslant 20$. There is an algorithm that, for all but at most $q^{-\Omega(nm)}$ fraction of m-dimensional alternating matrix spaces $\mathcal{A}$ in $\Lambda(n, q)$, tests the isometry of $\mathcal{A}$ to an arbitrary m-dimensional alternating matrix space $\mathcal{B}$, in time $q^{O(n+m)}$.*

The algorithm in Theorem 1 significantly improves over the Li-Qiao algorithm:

- First, it removes the linear dependence of m on n. The Li-Qiao algorithm inherently requires this linear dependence. That is, for general m, the Li-Qiao algorithm does *not* run in time $q^{O(n+m)}$. Indeed, some of its techniques do not work when m is even $\Omega(n^{1+\epsilon})$ or $O(n^{1-\epsilon})$ with some $0 < \epsilon < 1$.
- Second, even in the case of $m = \Theta(n)$, the average-case analysis of our algorithm is better: it works for all but $q^{-\Omega(n^2)}$ fraction of $\mathcal{A}$, while the Li-Qiao algorithm works for all but $q^{-\Omega(n)}$ fraction.
- Third, our algorithm is considerably simpler than the Li-Qiao algorithm, as the reader may compare in the descriptions of these algorithms in Subsection 1.2 and Subsection 1.3.

Partly because of the simplicity of our algorithm, we implement our algorithm, with suitable modifications and some heuristic shortcuts, in Magma [6]. Our experiments indicate that it improves significantly over default (brute-force) algorithms for this problem.

An immediate open problem is to examine whether our isometry testing algorithm can be strengthened to a canonical form algorithm for random alternating matrix spaces. For graph isomorphism, efficient canonical form algorithms for random graphs have long been known [2, 3]. However, to the best of authors' knowledge, there have been no canonical form algorithms for random alternating matrix spaces even in time $q^{o(n^2)}$.

1.1 Motivation and related works

Group isomorphism problem. The main motivation for us to study the AltMatSpIso problem is to understand the complexity of the Group Isomorphism (GpI) problem: Deciding whether two finite groups of order N are isomorphic. The complexities of GpI depend on how groups are represented in algorithms. When groups are specified by their multiplication (Cayley) tables, GpI reduces to the Graph Isomorphism problem (GI); cf. [24, Section 10]. When groups are represented by generators as permutations, matrices, or black-box groups, GI reduces to GpI; cf. [14, 23, 24, 13]. In either input model, the state-of-the-art algorithm runs in time *quasi-polynomial* in the group order [11, 26].

For our purpose, we shall mostly focus on the Cayley table model, since we do not even know an $N^{o(\log N)}$-time algorithm [29] (log to the base 2), despite that a simple $N^{\log N+O(1)}$-time algorithm has been known for decades [11, 26]. We note that Rosenbaum presented an algorithm in time $N^{\frac{1}{2}\log N+O(1)}$ [27]. Moreover, following Babai's breakthrough proof that GI is in quasi-polynomial time [1, 15], GpI in Cayley table model is now a key bottleneck to put GI in P, as Babai himself pointed out [1, Sec. 13.2 & 13.4 in arXiv version]. The past few years have witnessed a resurgence of activity on algorithms for this problem with worst-case analyses in terms of the group order. We refer the reader to [12] which contains a survey of these algorithms.

A natural approach to tackle GpI is to assume our given groups lie in a certain group class. For instance, for Abelian groups, one can test their isomorphism in linear time [19]. However, moving out Abelian a little bit, p-groups of class 2 and exponent p, the next natural group class beyond Abelian groups, pose great difficulty. Recall that a group is of exponent p if any nontrivial element has order p, and a group is of class 2 if its commutator subgroup is contained in its centre. Recent works [20, 9, 7, 21] solved some nontrivial subclasses, and lead to substantial improvement in practical algorithms. But their methods do not lead to any improvement for the worst-case time complexity of the general class.

***p*-groups of class 2 and exponent *p*, and alternating matrix spaces.** Alternating matrix spaces naturally appear in the study of p-groups of class 2 and exponent p via Baer's correspondence [4] for a prime $p > 2$. In fact, because of this correspondence, most recent works on this class of groups study alternating matrix spaces or alternating bilinear maps [20, 9, 7, 21]. We review this correspondence briefly. Given such a p-group, by taking the commutator map, one obtains an alternating bilinear map. Conversely, given an alternating bilinear map, one can construct such a p-group using an explicit formula (see e.g. [12, Fact 4.3]). Given an alternating bilinear map $\mathbb{F}_p^n \times \mathbb{F}_p^n \to \mathbb{F}_p^m$, one can obtain m $n \times n$ alternating matrices representing it, and take the linear span to get an alternating matrix space. Given an alternating matrix space, by taking a linear basis, one can obtain an alternating bilinear map. The above procedures preserve and respect isomorphism types of groups and isometry types of alternating matrix spaces. Moreover, it is easy to verify that the above procedures are computationally efficient in the Cayley table model [1]. Therefore, testing isomorphism of p-groups of class 2 and exponent p in time polynomial in the group order reduces to solving AltMatSpIso over $\mathbb{F}_p$ in time $p^{O(n+m)}$. Because of the current status of GpI, we see that solving AltMatSpIso in $q^{O(n+m)}$ is already very difficult. Recently, it has been shown that a large number of (hard) isomorphism problems can be reduced to AltMatSpIso in polynomial time; we refer the readers to [13] for more details. As an application, in [18], one can also build post-quantum cryptography schemes based on AltMatSpIso.

Current status of AltMatSpIso. For the AltMatSpIso problem, the brute-force algorithm takes time $q^{\Theta(n^2)} \cdot \mathrm{poly}(n, m, \log q)$. There are two slightly improved worst-case algorithms within certain range of parameters: the $N^{\frac{1}{4}\log_p N+O(1)}$-time algorithm for p-group isomorphism by Rosenbaum [27] and Rosenbaum and Wagner [28] translates to a $p^{\frac{1}{4}(n+m)^2+O(n+m)}$-time algorithm for AltMatSpIso over $\mathbb{F}_p$. Li and Qiao [21] adapted Luks' dynamic programming technique for GI [25] to obtain a $q^{\frac{1}{4}(n^2+m^2)+O(n+m)}$-time algorithm for AltMatSpIso. Note that both algorithms only improve the worst-case time-complexity when $m \leqslant n$; in fact, if $m = \Omega(n^2)$, then the brute-force algorithm already runs in time $q^{O(n+m)}$.

Another extreme case is when $m = O(1)$, where AltMatSpIso can be solved in $\mathrm{poly}(n, q)$ time for odd q. The algorithm is based on reducing AltMatSpIso to the Alternating Matrix Tuple Isometry (AltMatTupIso) problem, which asks the following: Given two m-tuples of alternating matrices $\mathbf{A}, \mathbf{B} \in \Lambda(n, q)^m$, decide whether there is an invertible matrix $T \in \mathrm{GL}(n, q)$ such that $T^{\mathrm{t}}\mathbf{A}T = (T^{\mathrm{t}}A_1T, \ldots, T^{\mathrm{t}}A_mT) = (B_1, \ldots, B_m) = \mathbf{B}$. Unlike AltMatSpIso, the AltMatTupIso problem can be solved in time $\mathrm{poly}(n, m, \log q)$ for *odd* q [17]. The reduction from AltMatSpIso to AltMatTupIso is by fixing a tuple of alternating matrices $\mathbf{A}$ in $\mathcal{A}$ as basis, and enumerate all possible m-tuples of alternating matrices $\mathbf{B}$ in $\mathcal{B}^m$. Then one can invoke the algorithm for AltMatTupIso to test isometry between $\mathbf{A}$ and $\mathbf{B}$ efficiently.

1.2 Average-case algorithms for GI and AltMatSpIso

In this subsection, we review the average-case algorithms for GI [2] and for AltMatSpIso [21].

To obtain average-case algorithms for GI (resp. AltMatSpIso), one needs to identify a property which is satisfied by almost all graphs (resp. alternating matrix spaces) chosen from a certain random model. Then for those graphs (resp. AltMatSpIso) satisfying the property, we test its isomorphism (resp. isometry) with an arbitrary graph (resp. alternating matrix space). The efficiency of the algorithm is guaranteed by both the efficiency of testing the property and the efficiency of the isomorphism (resp. isometry) testing.

We clarify the random models. In the graph setting, a natural choice is the celebrated Erdős-Rényi model [10], in which an n-vertex m-edge graph is endowed with probability $\binom{\binom{n}{2}}{m}^{-1}$. In the alternating matrix space setting, Li and Qiao defined a linear algebraic analogue of the Erdős-Rényi model, where each m-dimensional alternating matrix space in $\Lambda(n, q)$ will be chosen with probability $\begin{bmatrix}\binom{n}{2}\\ m\end{bmatrix}_q^{-1}$; see also Definition 2.

Average-case algorithm for GI. We first define a property $\mathcal{P}$ for graphs, which is a variant used in [2]. Let $G = ([n], E)$ be a simple undirected graph. Let $r \leqslant n$ be a positive integer, $S = [r]$ and $S' = [n] \setminus [r]$. Let $B = (S \cup S', F)$ be the bipartite graph induced by the cut (S, S') in G, where $F = \{\{i, j\} \in E : \ i \in S, \ j \in S'\}$. We label each $j \in S'$ by its adjacency

[1] The procedures are even efficient in matrix groups over finite fields [13, Lemma 7.5].

relations with those vertices in S. That is, assign an r-bit string $f_j \in \{0,1\}^r$ to each $j \in S'$ such that $f_j(i) = 1$ if and only if $\{i,j\} \in F$. We say a graph G satisfies property $\mathcal{P}$ if f_j's are distinct over $j \in S'$. It is easy to verify that, choosing $r = \lceil 3 \log n \rceil$, all but at most $n^{-\Omega(1)}$ fraction of graphs in the Erdős-Rényi model satisfy the property $\mathcal{P}$.

Here is an algorithm which tests isomorphism between graph G (satisfying property $\mathcal{P}$) and $H = ([n], E')$ (an arbitrary one), based on the well-known *individualisation and refinement* procedure. Let St_G be the set of r-bit strings obtained from the property $\mathcal{P}$. Note that $|\mathrm{St}_G| = n - r$. In the individualisation step, we enumerate all r-tuples of vertices in H. For each r-tuple $(i_1, \dots, i_r) \in [n]^r$, we perform the refinement step: assign the remaining vertices in H r-bit strings according to their adjacency relations with the r-tuple $(i_1, \dots, i_r)$ as before, to obtain another set of bit strings St_H. If $\mathrm{St}_G \neq \mathrm{St}_H$ we neglect this r-tuple; otherwise we obtained a bijective map from $[n]$ to $[n]$ by mapping j to i_j when $j \in [r]$ and the rest according to the labels. The last step is to check whether this bijective map induces an isomorphism between G and H.

The above algorithm runs in time $n^{O(\log n)}$ if G satisfies property $\mathcal{P}$. To recover the algorithm in [2], one can canonicalise the choice of the r-tuples by choosing the one with largest r degrees for both G and H, assuming that the largest r degrees are distinct. (This is another property which is satisfied by most graphs.)

Average-case algorithm for AltMatSpIso. Li and Qiao generalised the aforementioned graph property and the individualisation and refinement procedure to the alternating matrix space setting. It is helpful to think of alternating matrix spaces as a linear algebraic analogue of graphs. That is, we view vectors in $\mathbb{F}_q^n$ as a linear algebraic analogue of vertices. We then viewing alternating matrices as a linear algebraic analogue of edges. This is because we can think of edges as binary relations and alternating matrices as alternating bilinear forms on vectors.

We first define a property $\mathcal{Q}$ for alternating matrix spaces, in light of the one defined for graphs. Let $\mathcal{A}$ be an m-dimensional alternating matrix space in $\Lambda(n,q)$, and let $r \leqslant n$ be a positive integer. Let U_0 and V_0 be the $n \times r$ and $n \times (n-r)$ matrices over $\mathbb{F}_q$, whose columns are the first r and last $(n-r)$ standard basis, respectively. Let $\mathcal{A}_{U_0,V_0} = U_0{}^{\mathrm{t}} \mathcal{A} V_0 = \mathrm{span}\{U_0{}^{\mathrm{t}} A V_0 : A \in \mathcal{A}\}$, which is a subspace of $\mathrm{M}(r \times (n-r), q)$. The role of $\mathcal{A}_{U_0,V_0}$ is similar to the role of the bipartite graph $B = (S \cup S', F)$ in the graph setting.[2] Recall that in the graph setting, the hope was to label vertices in S' *uniquely*, meaning that they should have different adjacency relations with vertices in S. An equivalent way of saying this is that the right automorphism group of this bipartite graph B – those permutations of the right-hand-side vertices preserving the graph structure – is trivial. Inspired by this, Li and Qiao define the property $\mathcal{Q}$ on $\mathcal{A}$ as $\mathcal{R} := \{R \in \mathrm{GL}(n-r,q) : \mathcal{A}_{U_0,V_0} R = \mathcal{A}_{U_0,V_0}\}$ has size at most q^n, where $\mathcal{A}_{U_0,V_0} R = \{BR : B \in \mathcal{A}_{U_0,V_0}\}$. Here, $\mathcal{R}$ can be thought of as the corresponding to the right automorphism group in the graph setting. In [21], it is shown that when m and n are linearly related ($m = \Theta(n)$) and r is a constant (depending on the ratio m/n), $|\mathcal{R}| \leqslant q^n$ for all but at most $q^{-\Omega(n)}$ fractions of alternating matrix spaces.

Now we outline the Li-Qiao algorithm for isometry testing. Let $\mathcal{B}$ be another m-dimensional alternating matrix space in $\Lambda(n,q)$ for testing isometry with $\mathcal{A}$ (satisfying the property $\mathcal{Q}$). In the individualisation step, we enumerate all $n \times r$ matrices $U \in \mathrm{M}(n \times r, q)$ whose columns are linearly independent. Denote the columns space of U by C_U. We also need to enumerate all $(n-r)$-dimensional subspaces C_V such that $C_U \oplus C_V = \mathbb{F}_q^n$. C_V is specified

[2] Matrix spaces of the form $\mathcal{A}_{U_0,V_0}$ are studied as a linear algebraic analogue of cuts on graphs in [22].

by an $n \times (n-r)$ matrix V whose columns span C_V. Let $\mathcal{B}_{U,V} = \{U^{\mathrm{t}}BV : B \in \mathcal{B}\}$, which is a subspace of $\mathrm{M}(n \times (n-r), q)$. For each (U, V) with $\mathcal{B}_{U,V}$ we perform the refinement step. That is, we enumerate all $R \in \mathrm{GL}(n-r, q)$ such that $\mathcal{A}_{U_0,V_0}R = \mathcal{B}_{U,V}$, which can be done using algorithms from [8, 16][3]. If no such R exists we neglect the pair (U, V). Otherwise, we can recover an invertible $T \in \mathrm{GL}(n, q)$ from the information of (U, V) and R. Finally, check whether T is an isometry between $\mathcal{A}$ and $\mathcal{B}$.

The above algorithm runs in time $q^{O(n)}$, if $\mathcal{A}$ satisfies property $\mathcal{Q}$ (recall that n and m are linearly related and $r = O(1)$). In particular, the two enumerations in the individualisation step take time at most $q^{rn+r(n-r)} = q^{O(n)}$. In the refinement step, since $\mathcal{A}$ satisfies property $\mathcal{Q}$, $|\{R \in \mathrm{GL}(n-r, q) : \mathcal{A}_{U_0,V_0}R = \mathcal{B}_{U,V}\}|$ can be bounded above by q^n, as the set $\{R \in \mathrm{GL}(n-r, q) : \mathcal{A}_{U_0,V_0}R = \mathcal{B}_{U,V}\}$ is either empty, or a coset of the group H, whose order is upper bounded by q^n as $\mathcal{A}$ satisfies property $\mathcal{Q}$ we imposed. Thus the enumeration cost in the refinement step is at most q^n. All the other steps can be carried out in time $\mathrm{poly}(n, m, \log q)$; we refer the readers to [21] for more technical details on how to verify whether $\mathcal{A}$ satisfies the property $\mathcal{Q}$ and how to enumerate elements in $\{R \in \mathrm{GL}(n-r, q) : \mathcal{A}_{U_0,V_0}R = \mathcal{B}_{U,V}\}$ and get the invertible matrix T from R and (U, V).

1.3 A simplified algorithm and its implementation

Although it is a nice linear algebraic analogue of the algorithm in [2], the algorithm in [21] has several drawbacks. First, the algorithm only works when m and n are linear related. Second, the algorithm is still somewhat tricky, which makes it difficult to put into actual implementation. In this subsection, we describe a *simpler* average-case algorithm for AltMatSpIso, which works for all m and n (but only for odd q) and achieves the same performance as the one of Li–Qiao. A detailed description can be found in Subsection 4.1. This simplified algorithm already captures the essence of the strategy. The main algorithm for Theorem 1 in Subsection 4.3 further works for any q and achieves better average-case analysis.

The key idea behind our algorithm. Let $\mathcal{A} \leqslant \Lambda(n, q)$. The key idea behind our algorithm is to replace individualising r-dimensional subspaces of $\mathbb{F}_q^n$ by individualising r-dimensional subspaces of $\mathcal{A}$. This is inspired by the notion of *genus* of p-groups of class 2 and exponent p in [7].

Roughly speaking, genus-r p-groups of class 2 and exponent p correspond to r-dimensional alternating matrix spaces over $\mathbb{F}_p$. In [7], it is shown that even genus-2 p-groups of class 2 and exponent p demonstrate interesting behaviours. That is to say, constant-genus p-groups are already non-trivial objects. This leads us to consider individualising constant-dimensional subspaces of $\mathcal{A}$. In the graph setting, this would correspond to individualising r edges, which does not differ much from individualising $2r$ vertices, as each edge connects to two vertices. In the alternating matrix space setting, it turns out that individualising r-dimensional subspaces of $\mathcal{A}$ could impose severe constraints on the possible isometries, if this subspace satisfies certain generic conditions. This is not so surprising, as one full-rank alternating matrix is much more "powerful" than a single edge. Reflecting back, the combination of individualisation and refinement from graphs and the genus concept from p-groups reveals a nice interaction between graph-theoretic techniques and group-theoretic notions. We would

[3] In fact, the uses of [16, 8] are not necessary, as one can relax the property $\mathcal{Q}$ and then only use certain linear algebra computations; see [21].

also like to add that, the reason for the algorithm in [21] to individualising r-dimensional subspaces of $\mathbb{F}_q^n$ is because it was a close analogy of the average-case graph isomorphism algorithm of Babai, Erdős, and Selkow [2].

Algorithm outline. We first state another property $\mathcal{Q}'$ on a m-dimensional $\mathcal{A} \leqslant \Lambda(n,q)$. Let $\mathbf{A} = (A_1, \ldots, A_m)$ be a tuple of ordered linear basis of $\mathcal{A}$. For $c \in \mathbb{N}$, set $\mathbf{A}_c = (A_1, \ldots, A_c)$. We say $\mathcal{A}$ satisfies the property $\mathcal{Q}'$ if the group $\mathrm{Aut}(\mathbf{A}_c) := \{T \in \mathrm{GL}(n,q) : T^{\mathrm{t}}\mathbf{A}_c T = \mathbf{A}_c\}$ has order at most q^n.[4] When c is a large enough constant, all but at most $q^{-\Omega(n)}$ fraction of alternating matrix spaces satisfy property $\mathcal{Q}$, as shown in Theorem 5. Furthermore by algorithms in [9] (cf. Theorem 8), a generating set of $\mathrm{Aut}(\mathbf{A}_c)$ can be computed in time $\mathrm{poly}(n, \log q)$.

Our algorithm can be now stated as follows. Assume we would like to test isometry for $\mathcal{A}$ (satisfying the property $\mathcal{Q}'$) with an arbitrary $\mathcal{B}$, specified as two m-tuples of alternating matrices $\mathbf{A} = (A_1, \ldots, A_m)$ and $\mathbf{B} = (B_1, \ldots, B_m)$ from $\Lambda(n,q)^m$ for sufficiently large m and odd q. In the individualisation step, enumerate all c-tuples of alternating matrices $\mathbf{B}_c = (B_1, \ldots, B_c) \in \mathcal{B}^c$. For each $\mathbf{B}_c$ we perform the refinement step: Test isometry for alternating matrix tuples $\mathbf{A}_c$ and $\mathbf{B}_c$, using the $\mathrm{poly}(n, c, \log q)$-time algorithm in [17] (cf. Theorem 8). If they are not isometric, we neglect $\mathbf{B}_c$. Otherwise, the algorithm from [17] computes a specific isometry. Because we have computed $\mathrm{Aut}(\mathbf{A}_c)$, we can enumerate over all isometry T such that $T^{\mathrm{t}}\mathbf{A}_c T = \mathbf{B}_c$, and check whether $T^{\mathrm{t}}\mathcal{A}T = \mathcal{B}$.

The correctness lies in the simple fact that every isometry of $\mathcal{A}$ and $\mathcal{B}$ maps $\mathbf{A}_c$ to a c-tuple $\mathbf{B}_c$ in $\mathcal{B}^c$. The running time of the above procedure is $q^{O(n+m)}$ if $\mathcal{A}$ satisfies property $\mathcal{Q}'$. In particular, the enumeration cost in the individualisation step is at most q^{cm}. In the refinement step, enumerating all invertible matrix T such that $T^{\mathrm{t}}\mathbf{A}_c T = \mathbf{B}_c$ can be done in time $q^{O(n)} \cdot \mathrm{poly}(n, m, \log q)$ when q is odd, since $\{T \in \mathrm{GL}(n,q) : T^{\mathrm{t}}\mathbf{A}_c T = \mathbf{B}_c\}$ is a coset of the group $\{T \in \mathrm{GL}(n,q) : T^{\mathrm{t}}\mathbf{A}_c T = \mathbf{A}_c\}$, whose size is upper bounded by q^n by property $\mathcal{Q}'$. This algorithm is clearly simpler than the Li-Qiao algorithm, as it does not need to compute $\mathcal{B}_{U,V}$ etc..

Implementation and Performance. A bonus is our algorithm is more suitable to implement. We do so in Magma with some key adjustments, as detailed in Subsection 4.2. The implementation is publicly available on GitHub as part of a comprehensive collection of tools – developed and maintained by P. A. Brooksbank, J. Maglione, J. B. Wilson and their collaborators – to compute with groups, algebras, and multilinear functions [6].

The implementation is more convenient to describe using alternating bilinear maps (as the default algorithms do). A bilinear map $\alpha : V \times V \to W$ is alternating if for any $v \in V$, $\alpha(v,v) = 0$. Two alternating bilinear maps $\alpha, \beta : V \times V \to W$ are pseudo-isometric, if they are the same up to the natural action of $\mathrm{GL}(V) \times \mathrm{GL}(W)$. Let $V \cong \mathbb{F}_q^n$ and $W \cong \mathbb{F}_q^m$. An alternating bilinear map α can be specified by an m-tuple of alternating matrices from $\Lambda(n,q)$. Let α and β be represented by the alternating matrix tuples $\mathbf{A}$ and $\mathbf{B}$, respectively and $\mathcal{A}$ and $\mathcal{B}$ be the corresponding alternating matrix spaces of $\mathbf{A}$ and $\mathbf{B}$, respectively. It is readily verified that α and β are pseudo-isometric if and only if $\mathcal{A}$ and $\mathcal{B}$ are isometric.

Absent additional characteristic structure that can be exploited, the traditional (brute force) approach to deciding pseudo-isometry between alternating bilinear maps $\alpha, \beta \colon V \times V \to W$ is as follows. Let $\hat{\alpha}, \hat{\beta} \colon V \wedge V \to W$ denote the linear maps induced by α, β ($V \wedge V$ is the wedge product of a vector space V with itself). Compute the natural (diagonal) action of $\mathrm{GL}(V)$ on $V \wedge V$, and decide if $\ker \hat{\alpha}$ and $\ker \hat{\beta}$ – each of codimension $\dim W$ in $V \wedge V$ –

[4] The alert reader would note that the property defined here depends on the choice of bases of $\mathcal{A}$. This is not an essential problem due to the discussion in Section 3.

belong to the same orbit. An alternative version of brute force is to enumerate $\mathrm{GL}(W)$ and check if one of these transformations lifts to a pseudo-isometry from α to β. Which of these two brute-force options represents the best choice depends on the dimensions of V and W.

Our implementation is typically an improvement over both options. For example, in a preliminary experiment, our implementation readily decides pseudo-isometry between randomly selected alternating bilinear maps $\mathbb{F}_3^5 \times \mathbb{F}_3^5 \to \mathbb{F}_3^4$, while both brute-force options failed to complete. Note that the worst-case for all methods should be when α, β are *not* isometric, since in that case one must exhaust the entire enumerated list (or orbit) to confirm non-equivalence. However, the modifications we made tend to detect non-equivalence rather easily, since other (easily computed) invariants typically do not align in this case. We were therefore careful to also run tests with equivalent inputs, so as to ensure a fair comparison with default methods.

2 Preliminaries

Let $[m] = \{1, \dots, m\}$ for $m \in \mathbb{N}$. Let $\begin{bmatrix} n \\ d \end{bmatrix}_q = \frac{(1-q^n)\cdot \dots \cdot(1-q^{n-d+1})}{(1-q^d)\cdot \dots \cdot(1-q)}$ denote the Gaussian binomial coefficient with parameters n, d and base q. Note that $\begin{bmatrix} n \\ d \end{bmatrix}_q$ counts the number of d-dimensional subspaces of $\mathbb{F}_q^n$. The bound $\begin{bmatrix} n \\ d \end{bmatrix}_q \leqslant q^{nd}$ is useful.

Let $\mathrm{M}(n \times n', \mathbb{F})$ (resp. $\mathrm{M}(n, \mathbb{F})$) be the linear space of all $n \times n'$ (resp. $n \times n$) matrices over $\mathbb{F}$. For a matrix $A \in \mathrm{M}(n \times n', \mathbb{F})$, $A^{\mathrm{t}} \in \mathrm{M}(n' \times n, \mathbb{F})$ denotes the transpose of A. We use $A(i, j)$ to denotes the (i, j)th entry of the matrix A. The general linear group of degree n over $\mathbb{F}$ is denoted by $\mathrm{GL}(n, \mathbb{F})$. When $\mathbb{F} = \mathbb{F}_q$ for some prime power q, we write simply $\mathrm{M}(n, q)$ and $\mathrm{GL}(n, q)$ in place of $\mathrm{M}(n, \mathbb{F}_q)$ and $\mathrm{GL}(n, \mathbb{F}_q)$. An $n \times n$ matrix A over $\mathbb{F}$ is alternating if for every $v \in \mathbb{F}^n$, $v^{\mathrm{t}}Av = 0$. When $\mathbb{F}$ is not of characteristic 2, this is equivalent to the skew-symmetric condition. Let $\Lambda(n, \mathbb{F})$ be the linear space of all $n \times n$ alternating matrices over $\mathbb{F}$ (and $\Lambda(n, q)$ when $\mathbb{F} = \mathbb{F}_q$). We denote $\mathbf{A} = (A_1, \dots, A_m) \in \Lambda(n, q)^m$ to be an alternating matrix tuple and $\mathcal{A} = \mathrm{span}\{A_1, \dots, A_m\} \leqslant \Lambda(n, q)$ be an (the corresponding) alternating matrix space ($\leqslant$ denote the subspace notation).

We say two alternating matrix tuples $\mathbf{A}, \mathbf{B} \in \Lambda(n, \mathbb{F})^m$ are *isometric* if there exists $T \in \mathrm{GL}(n, \mathbb{F})$ such that

$$T^{\mathrm{t}}\mathbf{A}T := (T^{\mathrm{t}}A_1T, \dots, T^{\mathrm{t}}A_mT) = (B_1, \dots, B_m) = \mathbf{B}.$$

The set of isometries between $\mathbf{A}$ and $\mathbf{B}$ is denoted as

$$\mathrm{Isom}(\mathbf{A}, \mathbf{B}) = \{T \in \mathrm{GL}(n, \mathbb{F}) : T^{\mathrm{t}}\mathbf{A}T = \mathbf{B}\};$$

the group of *autometries* (or self-isometries) of $\mathbf{A}$ is denoted as $\mathrm{Aut}(\mathbf{A}) = \mathrm{Isom}(\mathbf{A}, \mathbf{A})$. We say two alternating matrix tuples $\mathbf{A}, \mathbf{B} \in \Lambda(n, \mathbb{F})^m$ are *pseudo-isometric* if there exist $T \in \mathrm{GL}(n, \mathbb{F})$ and $R \in \mathrm{GL}(m, \mathbb{F})$ such that

$$T^{\mathrm{t}}\mathbf{A}T = (T^{\mathrm{t}}A_1T, \dots, T^{\mathrm{t}}A_mT) = (\sum_{j=1}^{m} R(1, j)B_j, \dots, \sum_{j=1}^{m} R(m, j)B_j) =: \mathbf{B}^R.$$

The set of pseudo-isometries between $\mathbf{A}$ and $\mathbf{B}$ is defined as

$$\Psi\mathrm{Isom}(\mathbf{A}, \mathbf{B}) = \{T \in \mathrm{GL}(n, \mathbb{F}) : \exists\ R \in \mathrm{GL}(m, q),\ T^{\mathrm{t}}\mathbf{A}T = \mathbf{B}^R\}.$$

The group of *pseudo-autometries* (or self-pseudo-isometries) of $\mathbf{A}$ is denoted as $\Psi\mathrm{Aut}(\mathbf{A}) = \Psi\mathrm{Isom}(\mathbf{A}, \mathbf{A})$. It is straightforward to see that $\mathrm{Isom}(\mathbf{A}, \mathbf{B})$ (resp. $\Psi\mathrm{Isom}(\mathbf{A}, \mathbf{B})$) is a (possibly empty) coset of $\mathrm{Aut}(\mathbf{A})$ (resp. $\Psi\mathrm{Aut}(\mathbf{A})$).

We say two alternating matrix spaces $\mathcal{A}, \mathcal{B} \leqslant \Lambda(n, \mathbb{F})$ are *isometric*, if there exists $T \in \mathrm{GL}(n, \mathbb{F})$ such that $T^{\mathrm{t}} \mathcal{A} T := \{T^{\mathrm{t}} A T : A \in \mathcal{A}\} = \mathcal{B}$. We can define the coset of isometries from $\mathcal{A}$ to $\mathcal{B}$ $\mathrm{Isom}(\mathcal{A}, \mathcal{B})$, and the group of autometries $\mathrm{Aut}(\mathcal{A})$, for alternating matrix spaces. If $\mathcal{A}$ and $\mathcal{B}$ are the corresponding alternating matrix spaces spanned by $\mathbf{A}$ and $\mathbf{B}$, respectively, then $\mathcal{A}$ and $\mathcal{B}$ are isometric if and only if $\mathbf{A}$ and $\mathbf{B}$ are pseudo-isometric, i.e. $\mathrm{Isom}(\mathcal{A}, \mathcal{B}) = \Psi\mathrm{Isom}(\mathbf{A}, \mathbf{B})$.

For two tuples of alternating matrices $\mathbf{A}, \mathbf{B} \in \Lambda(n, \mathbb{F})^m$, the the *adjoint space* from $\mathbf{A}$ to $\mathbf{B}$ is defined as $\mathrm{Adj}(\mathbf{A}, \mathbf{B}) = \{(T, T') \in \mathrm{M}(n, \mathbb{F}) \oplus \mathrm{M}(n, \mathbb{F}) : T\mathbf{A} = \mathbf{B}T'\}$. The *adjoint algebra* of $\mathbf{A}$ is defined as $\mathrm{Adj}(\mathbf{A}) = \{(T, T') \in \mathrm{M}(n, \mathbb{F}) \oplus \mathrm{M}(n, \mathbb{F}) : T\mathbf{A} = \mathbf{A}T'\}$. Clearly, if $T \in \mathrm{Aut}(\mathbf{A})$, then $(T^{\mathrm{t}}, T^{-1}) \in \mathrm{Adj}(\mathbf{A})$. Furthermore, if $\mathbf{A}$ and $\mathbf{B}$ are isometric, then $|\mathrm{Adj}(\mathbf{A}, \mathbf{B})| = |\mathrm{Adj}(\mathbf{A})|$.

3 Random models and average-case properties

We now formally define the linear algebraic analogue of the Erdős-Rényi model, which has been mentioned frequently in Section 1.

▶ **Definition 2** (The linear algebraic analogue of the Erdős-Rényi model). *The linear algebraic analogue of the Erdős-Rényi model,* $\mathrm{LinER}(n, m, q)$*, is the uniform probability distribution over the set of* m*-dimensional subspaces of* $\Lambda(n, q)$*, that is, each subspace is endowed with probability* $\begin{bmatrix} \binom{n}{2} \\ m \end{bmatrix}_q^{-1}$.

We also recall a random model for alternating matrix tuples, introduced in [21].

▶ **Definition 3** (The naive model for alternating matrix tuples). *The naive model for alternating matrix tuples,* $\mathrm{NaiT}(n, m, q)$*, is the probability distribution over the set of all* m*-tuples of* $n \times n$ *alternating matrices over* $\mathbb{F}_q$*, where each tuple is endowed with probability* $q^{-\binom{n}{2}m}$.

A useful fact is that if we would like to show a certain property holds with high probability for alternating matrix spaces in $\mathrm{LinER}(n, m, q)$, we can in turn show that a corresponding property holds with high probability for alternating matrix tuples in $\mathrm{NaiT}(n, m, q)$. The statement can be quantified as follows: Suppose we have $\mathcal{P}(n, m, q)$, a property of m-dimensional alternating matrix spaces in $\Lambda(n, q)$, and wish to show that $\mathcal{P}(n, m, q)$ holds with high probability in $\mathrm{LinER}(n, m, q)$. $\mathcal{P}(n, m, q)$ naturally induces $\mathcal{P}'(n, m, q)$, a property of alternating matrix tuples in $\Lambda(n, q)^m$ that span m-dimensional alternating matrix spaces. Let $\mathcal{Q}(n, m, q)$ be a property of all alternating matrix tuples in $\Lambda(n, q)^m$, so that $\mathcal{Q}(n, m, q)$ and $\mathcal{P}'(n, m, q)$ coincide when restricting to those alternating tuples spanning m matrix spaces. The following is proved in [21].

▶ **Proposition 4** ([21, Proposition 13 in arXiv version]). *Let* $\mathcal{P}(n, m, q)$ *and* $\mathcal{Q}(n, m, q)$ *be as above. Suppose in* $\mathrm{NaiT}(n, m, q)$*,* $\mathcal{Q}(n, m, q)$ *happens with probability at least* $1 - f(n, m, q)$ *for* $0 \leqslant f(n, m, q) < 1$*. Then in* $\mathrm{LinER}(n, m, q)$*,* $\mathcal{P}(n, m, q)$ *happens with probability at least* $1 - 4f(n, m, q)$*.*

In the rest of this paper, we assume a random alternating matrix space is chosen from the linear algebraic analogue of the Erdős-Rényi model and a random alternating matrix tuple is chosen from the naive model. To prove Theorem 1, it is sufficient to work with alternating matrix tuples and the naive model.

We now present the average-case property, which will be used in our algorithm. Recall that in Subsection 1.3, the desired property is to show that, for a random alternating matrix tuple $\mathbf{A} = (A_1, \ldots, A_m) \in \Lambda(n, q)^m$ and some $c = O(1)$, the tuple $\mathbf{A}_c = (A_1, \ldots, A_c)$ has autometry group $\mathrm{Aut}(\mathbf{A}_c)$ of size at most q^n with probability $1 - q^{-\Omega(n)}$. We prove a stronger statement.

▶ **Theorem 5.** *Let $c = 20$. For all but at most $q^{-\Omega(n)}$ fraction of $\mathbf{A}_c = (A_1, \dots, A_c) \in \Lambda(n, q)^c$, we have $|\operatorname{Adj}(\mathbf{A}_c)| \leqslant q^n$.*

Note that $|\operatorname{Aut}(\mathbf{A}_c)| \leqslant |\operatorname{Adj}(\mathbf{A}_c)|$ for any $\mathbf{A}_c$. To prove Theorem 5, we need the following from [21]. Given a tuple of matrices $\mathbf{A} = (A_1, \dots, A_r) \in \mathrm{M}(n, q)^r$, define the image of $U \leqslant \mathbb{F}_q^n$ under $\mathbf{A}$ as $\mathbf{A}(U) := \operatorname{span}\{A_i u : \ i \in [r], \ u \in U\}$.

▶ **Definition 6.** We say $\mathbf{A} = (A_1, \dots, A_r) \in \mathrm{M}(n, q)^r$ is *stable*, if for any nonempty proper subspace $U \leqslant \mathbb{F}_q^n$, we have $\dim(\mathbf{A}(U)) > \dim(U)$.

▶ **Proposition 7** ([21, Proposition 10 in arXiv version]). *If $\mathbf{A} \in \mathrm{M}(n, q)^r$ is stable, then $|\operatorname{Adj}(\mathbf{A})| \leqslant q^n$.*

Thus, to prove Theorem 5, we need to upper bound the probability of a random alternating matrix tuple being *not stable*. The proof is somewhat similar to the one in [21], with one interesting observation: We can use some random alternating matrices in $\Lambda(n, q)$ to "mimick" a random matrix $\mathbb{M}(n, q)$. The detail of the proof can be found, e.g. in [5, Section 6.3].

4 Average-case algorithms for AltMatSpIso

4.1 The simplified main algorithm

As we have mentioned in Subsection 1.3, our algorithm invokes the algorithm for testing isometry for alternating matrix tuples as subroutines, which is formally state it here.

▶ **Theorem 8** ([9, 17]). *Let $\mathbf{A}, \mathbf{B} \in \Lambda(n, q)^m$ for some odd q. There exists a $\operatorname{poly}(n, m, \log q)$-time algorithm which takes $\mathbf{A}$ and $\mathbf{B}$ as inputs and outputs $\operatorname{Isom}(\mathbf{A}, \mathbf{B})$, specified by (if nonempty) a generating set of $\operatorname{Aut}(\mathbf{A})$ (by the algorithm in [9]) and a coset representative $T \in \operatorname{Isom}(\mathbf{A}, \mathbf{B})$ (by the algorithm in [17]).*

We also need the following observation to enumerate elements in $\operatorname{Aut}(\mathbf{A})$, which follows easily by computing the closure of the given generating set.

▶ **Observation 9.** *Let $C_1, \dots, C_t \in \mathrm{GL}(n, q)$, and let G be the group generated by C_i's. Let $s \in \mathbb{N}$. Then there exists an algorithm that either reports that $|G| > s$, or lists all elements in G, in time $\operatorname{poly}(s, n, \log q)$.*

Now we formally describe the simplified main algorithm for AltMatSpIso stated in Subsection 1.3, that is Algorithm 1. Note that we are given alternating matrix tuples $\mathbf{A}$ and $\mathbf{B}$, which span $\mathcal{A}$ and $\mathcal{B}$, respectively. By the discussion in Section 2, we can equivalently decide the pseudo-isometry between $\mathbf{A}$ and $\mathbf{B}$.

▶ **Proposition 10.** *Algorithm 1 runs in time $\operatorname{poly}(q^{cm}, s, n)$.*

Proof. If Algorithm 1 outputs $|\operatorname{Aut}(\mathbf{A}_c)| > s$, then its running time is determined by Theorem 8 and Observation 9, which together require $\operatorname{poly}(s, n, \log q)$.

If $|\operatorname{Aut}(\mathbf{A}_c)| \leqslant s$, we analyse the two For-loops at Step 4 and Step 4c, respectively. The first loop adds a multiplicative factor of q^{cm}, since enumerating all matrices in $\mathcal{B}$ costs q^m. The second loop adds a multiplicative factor of $\operatorname{poly}(n, s, \log q)$, due to the fact that $|\operatorname{Isom}(\mathbf{A}_c, \mathbf{B}_c)| = |\operatorname{Aut}(\mathbf{A}_c)| \leqslant s$ (as $\operatorname{Isom}(\mathbf{A}_c, \mathbf{B}_c)$ is a coset of $\operatorname{Aut}(\mathbf{A}_c)$). Other steps can be carried out in time $\operatorname{poly}(n, \log q)$. Therefore the overall running time is upper bounded by $\operatorname{poly}(q^{cm}, s, n)$. ◀

We prove the correctness of Algorithm 1, if it does not report $|\operatorname{Aut}(\mathbf{A}_c)| > s$.

Algorithm 1 The first average-case algorithm for AltMatSpIso.

```
Input: $\mathbf{A} = (A_1, \dots, A_m), \mathbf{B} = (B_1, \dots, B_m) \in \Lambda(n, q)^m$, $c, s \in \mathbb{N}$, and $q$ is odd.
Output: Either (1) "$|\operatorname{Aut}(\mathbf{A}_c)| > s$.", or (2) $\Psi\mathrm{Isom}(\mathbf{A}, \mathbf{B})$ as a set $L$, which may be empty.
Algorithm procedure:
  1. Set $L \leftarrow \{\}$. Set $\mathbf{A}_c = (A_1, \dots, A_c)$.
  2. Use Theorem 8 to compute a generating set for $\operatorname{Aut}(\mathbf{A}_c)$.
  3. Use Observation 9 with input $s$ and the generating set of $\operatorname{Aut}(\mathbf{A}_c)$. (If $|\operatorname{Aut}(\mathbf{A}_c)| > s$,
     we terminate the algorithm and report that "$|\operatorname{Aut}(\mathbf{A}_c)| > s$.")
  4. Set $\mathcal{B} = \operatorname{span}\{B_1, \dots, B_m\}$; for every $\mathbf{B}_c = (B_1', \dots, B_c') \in \mathcal{B}^c$, do the following.
     (a) Use Theorem 8 to decide whether $\mathbf{A}_c$ and $\mathbf{B}_c$ are isometric.
     (b) If not, go to the next $\mathbf{B}_c$. Otherwise, we get the non-empty coset $\mathrm{Isom}(\mathbf{A}_c, \mathbf{B}_c)$.
     (c) For every $T \in \mathrm{Isom}(\mathbf{A}_c, \mathbf{B}_c)$, do the following.
           Test whether the linear spans of $T^{\mathrm{t}}\mathbf{A}T$ and $\mathbf{B}$ are the same. If not, go to the
           next $T$. If so, add $T$ into $L$.
  5. Output $L$.
```

▶ **Proposition 11.** *If Algorithm 1 does not report $|\operatorname{Aut}(\mathbf{A}_c)| > s$, then it lists the set of pseudo-isometries (resp. isometries) between $\mathbf{A}$ (resp. $\mathcal{A}$) and $\mathbf{B}$ (resp. $\mathcal{B}$). In particular, $|\mathrm{Isom}(\mathcal{A}, \mathcal{B})| = |\Psi\mathrm{Isom}(\mathbf{A}, \mathbf{B})| \leqslant q^{cm} \cdot s$.*

Proof. By Step 4c, every T added to L is a pseudo-isometry. We are left to show that L contains all the pseudo-isometries. For this, take any pseudo-isometry T. Since $T^{\mathrm{t}}\mathcal{A}T = \mathcal{B}$, we know $T^{\mathrm{t}}\mathbf{A}_c T$ is equal to some $\mathbf{B}_c \in \mathcal{B}^c$. So when enumerating these $\mathbf{B}_c$ in Step 4, T will pass all the tests in the following, and then be added to L. ◀

It remains to specify the choices of c and s in Algorithm 1. This can be done by Theorem 5.

▶ **Proposition 12.** *Let $c = 20$ and $s = q^n$. For all but at most $q^{-\Omega(n)}$ fraction of $\mathbf{A}_c = (A_1, \dots, A_c) \in \Lambda(n, q)^c$, we have $|\operatorname{Aut}(\mathbf{A}_c)| \leqslant s$.*

Proof. This is because, if $T \in \operatorname{Aut}(\mathbf{A}_c)$, then $(T^{\mathrm{t}}, T^{-1}) \in \operatorname{Adj}(\mathbf{A}_c)$. So $|\operatorname{Aut}(\mathbf{A}_c)| \leqslant |\operatorname{Adj}(\mathbf{A}_c)|$. ◀

Combining Propositions 10 to 12, we have the following theorem.

▶ **Theorem 13.** *Let $m \geqslant 20$, and let $\mathbb{F}_q$ be a finite field of odd size. For all but at most $q^{-\Omega(n)}$ fraction of $\mathbf{A} = (A_1, \dots, A_m) \in \Lambda(n, q)^m$, Algorithm 1 tests the isometry of $\mathbf{A}$ with an arbitrary $\mathbf{B} \in \Lambda(n, q)^m$ in time $q^{O(n+m)}$.*

4.2 Magma implementation of Algorithm 1

To make this algorithm suitable for practical purposes, recall that the algorithm's running time is dominated by the two For-loops which give multiplicative factors of q^{cm} and s, respectively. For the average-case analysis we used $c = 20$, but having this standing on the exponent is too expensive. In practice, actually using $c = 3$ already imposes a severe restriction on s, the order of $\operatorname{Aut}(\mathbf{A}_c)$. So we use $c = 3$ in the implementation which gives a reasonable performance.

But having q^{3m} in the For-loop is still too demanding. Indeed, in practice the tolerable enumeration is around 5^{10}, namely $q = 5$ and 10 on the exponent. So with $c = 3$, the range of m is still severely limited. (Interestingly, the algorithm seems to have a better dependence on n.) It is most desirable if we could let $c = 1$, namely simply q^m.

To achieve that we use the following heuristic. Note that if $A_1, \dots, A_c$ are low-rank matrices, then we will only need to match them with the low-rank matrices from $\mathcal{B}$. Our experiement shows that, for a random m-tuple of alternating matrices $\mathbf{A}$ over $\mathbb{F}_q$, when q is a small constant, the number of low-rank (i.e. non-full-rank) matrices in the linear span of $\mathbf{A}$ is expected to be small (i.e. much smaller than q^m) and non-zero (i.e. no less than 3) at the same time. Note that the set of all low-rank matrices can be computed in time $q^m \cdot \mathrm{poly}(n, \log q)$-time. We then choose 3 low-rank matrices from the linear span of $\mathbf{A}$. Then use $q^m \cdot \mathrm{poly}(n, \log q)$-time to compute the set of low-rank matrices from $\mathcal{B}$, denoted as $\mathcal{B}_l$. We can then replace enumerating $\mathcal{B}^c$ with $\mathcal{B}_l^c$, which in general is much smaller.

4.3 The main algorithm and proof of Theorem 1

We now introduce the algorithm (see Algorithm 2) that supports Theorem 1, which differs from Algorithm 1 in two places.

1. The first and major difference is to replace the uses of $\mathrm{Aut}(\mathbf{A}_c)$ and $\mathrm{Isom}(\mathbf{A}_c, \mathbf{B}_c)$ with the adjoint algebra $\mathrm{Adj}(\mathbf{A}_c)$ and adjoint space $\mathrm{Adj}(\mathbf{A}_c, \mathbf{B}_c)$, thereby avoiding using Theorem 8. Since $\mathrm{Adj}(\mathbf{A}_c)$ and $\mathrm{Adj}(\mathbf{A}_c, \mathbf{B}_c)$ are easy to compute over any field, this removes the odd q issue. On the other hand, although $\mathrm{Adj}(\mathbf{A}_c)$ and $\mathrm{Adj}(\mathbf{A}_c, \mathbf{B}_c)$ are also easier to analyse, $\mathrm{Adj}(\mathbf{A}_c)$ and $\mathrm{Adj}(\mathbf{A}_c, \mathbf{B}_c)$ could be larger than $\mathrm{Aut}(\mathbf{A}_c)$ and $\mathrm{Isom}(\mathbf{A}_c, \mathbf{B}_c)$, so they are less useful from the practical viewpoint.
2. The second place is step 2 in Algorithm 2: instead of just using the first c matrices as in the algorithm presented in Algorithm 1, Algorithm 2 slices the m matrices of $\mathbf{A}$ into $\lfloor m/c \rfloor$ segments of c-tuples of matrices, and tries each segment until it finds one segment with a small adjoint algebra. This step helps in improving the average-case analysis, and can be applied to the algorithm presented in Algorithm 1 as well.

► **Proposition 14.** *Algorithm 2 runs in time* $\mathrm{poly}(q^{cm}, s, n)$.

Proof. If Algorithm 2 outputs "$\mathbf{A}$ does not satisfy the generic condition.", then it just executes the For-loop in Step 2, which together runs in time $\mathrm{poly}(m, n, \log q)$.

Otherwise, there are two For-loops at Step 4 and Step 4c, which add multiplicative factors q^{cm} and s, respectively. Other steps can be carried out in time $\mathrm{poly}(n, \log q)$. Therefore the whole algorithm runs in time $\mathrm{poly}(q^{cm}, s, n)$. ◄

We prove the correctness of Algorithm 2 in the case that it does not report "$\mathbf{A}$ does not satisfy the generic condition."

► **Proposition 15.** *Suppose that Algorithm 2 does not report "$\mathbf{A}$ does not satisfy the generic condition." Then the algorithm lists the (possibly empty) set of pseudo-isometries between $\mathbf{A}$ and $\mathbf{B}$. In particular,* $|\Psi\mathrm{Isom}(\mathbf{A}, \mathbf{B})| \leqslant q^{cm} \cdot s$.

Proof. By Step 4c, every T^{t} added to L is a pseudo-isometry. So we are left to show that L contains all the pseudo-isometries. For this, take an arbitrary pseudo-isometry T. Then T sends $\mathbf{A}_c$ to some $\mathbf{B}_c \in \mathcal{B}^c$, i.e., $T^{\mathrm{t}}\mathbf{A}_c T = \mathbf{B}_c$. In particular, $(T^{\mathrm{t}}, T^{-1}) \in \mathrm{Adj}(\mathbf{A}_c, \mathbf{B}_c)$. So when enumerating this $\mathbf{B}_c \in \mathcal{B}^c$, (T^{t}, T^{-1}) will pass all the tests in the following, and T will be be added to L. ◄

Algorithm 2 The second average-case algorithm for AltMatSpIso.

```
Input: $\mathbf{A} = (A_1, \dots, A_m), \mathbf{B} = (B_1, \dots, B_m) \in \Lambda(n, q)^m$ and $c, s \in \mathbb{N}$.
Output: Either (1) "$\mathbf{A}$ does not satisfy the generic condition."; or (2) $\Psi\mathrm{Isom}(\mathbf{A}, \mathbf{B})$ as a set
    $L$, which may be empty.
Algorithm procedure:
  1. Set $L \leftarrow \{\}$. Set $F \leftarrow$ false.
  2. For $i = 1, \dots, \lfloor m/c \rfloor$, do the following.
    (a) Set $\mathbf{A}_c = (A_{c(i-1)+1}, \dots, A_{ci})$.
    (b) Compute a linear basis of $\mathrm{Adj}(\mathbf{A}_c) \subseteq \mathrm{M}(n, q) \oplus \mathrm{M}(n, q)$.
    (c) If $|\mathrm{Adj}(\mathbf{A}_c)| \leqslant s$, set $F$ to be true, and break the For-loop.
  3. If $F =$ false, return "$\mathbf{A}$ does not satisfy the generic condition." and terminate.
     Otherwise,
  4. Set $\mathcal{B} = \mathrm{span}\{B_1, \dots, B_m\}$; for every $\mathbf{B}_c = (B_1, \dots, B_c) \in \mathcal{B}^c$, do the following.
    (a) Compute a linear basis for $\mathrm{Adj}(\mathbf{A}_c, \mathbf{B}_c) \subseteq \mathrm{M}(n, q) \oplus \mathrm{M}(n, q)$.
    (b) If $|\mathrm{Adj}(\mathbf{A}_c, \mathbf{B}_c)| > s$, go to the next $\mathbf{B}_c$.
    (c) For every $(T, T') \in \mathrm{Adj}(\mathbf{A}_c, \mathbf{B}_c)$, do the following.
          If $T$ and $T'$ are invertible and $(T')^{-1} = T^{\mathrm{t}}$, test whether the linear spans of
          $T\mathbf{A}T^{\mathrm{t}}$ and $\mathbf{B}$ are the same. If not, go to the next $(T, T')$. If so, add $T^{\mathrm{t}}$ into $L$.
  5. Output $L$.
```

Now we specify the choice of $c = 20$ and $s = q^n$, based on Theorem 5.

▶ **Proposition 16.** *Let $m \geqslant c = 20$, and let $\ell = \lfloor m/c \rfloor \in \mathbb{N}$. For all but at most $q^{-\Omega(n\cdot\ell)} = q^{-\Omega(nm)}$ fraction of $\mathbf{A} = (A_1, \dots, A_m) \in \Lambda(n, q)^m$, there exists some $i \in [\ell]$, such that, letting $\mathbf{A}_{c,i} = (A_{c(i-1)+1}, \dots, A_{ci})$, we have $|\mathrm{Adj}(\mathbf{A}_{c,i})| \leqslant q^n$.*

Proof. We slice $\mathbf{A}$ into $\ell = \lfloor m/c \rfloor$ segments, where each segment consists of c random alternating matrices. Each segment is some $\mathbf{A}_{c,i} \in \Lambda(n, q)^c$, with $\Pr[|\mathrm{Adj}(\mathbf{A}_{c,i})| > q^n] \leqslant q^{-\Omega(n)}$ by Theorem 5. Since $A_1, \dots, A_m$ are chosen independently and uniformly at random, the probability of every $\mathbf{A}_{c,i} = (A_{c(i-1)+1}, \dots, A_{ci})$, $i \in [\ell]$, with $|\mathrm{Adj}(\mathbf{A}_{c,i})| > q^n$, is upper bounded by $(q^{-\Omega(n)})^\ell = q^{-\Omega(nm)}$. ◀

Theorem 1 then follows from Propositions 14 to 16.

References

1 László Babai. Graph Isomorphism in Quasipolynomial Time [extended abstract]. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, pages 684–697, 2016. arXiv:1512.03547v2. doi:10.1145/2897518.2897542.

2 László Babai, Paul Erdős, and Stanley M. Selkow. Random Graph Isomorphism. *SIAM Journal on Computing*, 9(3):628–635, 1980. doi:10.1137/0209047.

3 László Babai and Ludek Kucera. Canonical Labelling of Graphs in Linear Average Time. In *20th Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 29-31 October 1979*, pages 39–46, 1979. doi:10.1109/SFCS.1979.8.

4 Reinhold Baer. Groups with Abelian Central Quotient Group. *Transactions of the American Mathematical Society*, 44(3):357–386, 1938. doi:10.2307/1989886.

5 Peter A. Brooksbank, Joshua A. Grochow, Yinan Li, Youming Qiao, and James B. Wilson. Incorporating Weisfeiler-Leman into Algorithms for Group Isomorphism, 2019. arXiv. arXiv:1905.02518.

6 Peter A. Brooksbank, Joshua Maglione, and James B. Wilson. TheTensor.Space. https://github.com/thetensor-space/.

7 Peter A. Brooksbank, Joshua Maglione, and James B. Wilson. A Fast Isomorphism Test for Groups whose Lie Algebra has Genus 2. *Journal of Algebra*, 473:545–590, 2017. doi:10.1016/j.jalgebra.2016.12.007.

8 Peter A. Brooksbank and Eamonn A. O'Brien. Constructing the Group Preserving a System of Forms. *International Journal of Algebra and Computation*, 18(02):227–241, 2008. doi:10.1142/S021819670800441X.

9 Peter A. Brooksbank and James B. Wilson. Computing Isometry Groups of Hermitian Maps. *Transactions of the American Mathematical Society*, 364(4):1975–1996, 2012. doi:10.2307/41524909.

10 Paul Erdős and Alfréd Rényi. On Random Graphs I. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

11 Volkmar Felsch and Joachim Neubüser. On a Programme for the Determination of the Automorphism Group of a Finite Group. In *Computational Problems in Abstract Algebra*, pages 59–60. Pergamon, 1970. doi:10.1016/B978-0-08-012975-4.50011-4.

12 Joshua A. Grochow and Youming Qiao. Algorithms for Group Isomorphism via Group Extensions and Cohomology. *SIAM Journal on Computing*, 46(4):1153–1216, 2017. doi:10.1137/15M1009767.

13 Joshua A. Grochow and Youming Qiao. Isomorphism Problems for Tensors, Groups, and Cubic Forms: Completeness and Reductions, 2019. arXiv. arXiv:1907.00309.

14 Hermann Heineken and Hans Liebeck. The Occurrence of Finite Groups in the Automorphism Group of Nilpotent Groups of Class 2. *Archiv der Mathematik*, 25:8–16, 1974. doi:10.1007/BF01238631.

15 Harald Andrés Helfgott, Jitendra Bajpai, and Daniele Dona. Graph Isomorphisms in Quasi-polynomial Time, 2017. arXiv. arXiv:1710.04574.

16 Gábor Ivanyos, Marek Karpinski, and Nitin Saxena. Deterministic polynomial time algorithms for matrix completion problems. *SIAM Journal on Computing*, 39(8):3736–3751, 2010. doi:10.1137/090781231.

17 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 Journal on Computing*, 48(3):926–963, 2019. doi:10.1137/18M1165682.

18 Zhengfeng Ji, Youming Qiao, Fang Song, and Aaram Yun. General Linear Group Action on Tensors: A Candidate for Post-quantum Cryptography. In *Theory of Cryptography*, pages 251–281, 2019. doi:10.1007/978-3-030-36030-6_11.

19 Telikepalli Kavitha. Linear Time Algorithms for Abelian Group Isomorphism and Related Problems. *Journal of Computer and System Sciences*, 73(6):986–996, 2007. doi:10.1016/j.jcss.2007.03.013.

20 Mark L. Lewis and James B. Wilson. Isomorphism in Expanding Families of Indistinguishable Groups. *Groups Complexity Cryptology*, 4(1):73–110, 2012. doi:10.1515/gcc-2012-0008.

21 Yinan Li and Youming Qiao. Linear Algebraic Analogues of the Graph Isomorphism Problem and the Erdős-Rényi Model. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 463–474, 2017. arXiv:1708.04501v2. doi:10.1109/FOCS.2017.49.

22 Yinan Li and Youming Qiao. Group-theoretic Generalisations of Vertex and Edge Connectivities, 2019. arXiv. arXiv:1906.07948.

23 Ruvim Lipyanski and Natalia Vanetik. On Borel Complexity of the Isomorphism Problems for Graph related Classes of Lie Algebras and Finite p-groups. *Journal of Algebra and its Applications*, 14(5):1550078, 15, 2015. doi:10.1142/S0219498815500784.

24 Eugene M. Luks. Permutation Groups and Polynomial-time Computation. In *Groups and Computation*, volume 11 of *DIMACS: Series in Discrete Mathematics and Theoretical Computer Science*, 1993.

25 Eugene M. Luks. Hypergraph Isomorphism and Structural Equivalence of Boolean Functions. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing STOC*, pages 652–658. ACM, 1999. `doi:10.1145/301250.301427`.

26 Gary L. Miller. On the $n \log n$ Isomorphism Technique (A Preliminary Report). In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC '78, page 51–58, New York, NY, USA, 1978. Association for Computing Machinery. `doi:10.1145/800133.804331`.

27 David J. Rosenbaum. Bidirectional Collision Detection and Faster Deterministic Isomorphism Testing, 2013. arXiv. `arXiv:1304.3935`.

28 David J. Rosenbaum and Fabian Wagner. Beating the Generator-enumeration Bound for p-group Isomorphism. *Theoretical Computer Science*, 593:16–25, 2015. `doi:10.1016/j.tcs.2015.05.036`.

29 James B. Wilson. 2014 conference on *Groups, Computation, and Geometry* at Colorado State University, co-organized by P. Brooksbank, A. Hulpke, T. Penttila, J. Wilson, and W. Kantor. Personal communication, 2014.