58th Annual IEEE Symposium on Foundations of Computer Science

Linear algebraic analogues of the graph isomorphism problem and the Erdős-Rényi model

(Extended Abstract)

Yinan Li
Centre for Quantum Software and Information
University of Technology Sydney
Sydney, Australia
Yinan.Li@student.uts.edu.au

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

***Abstract*—A classical difficult isomorphism testing problem is to test isomorphism of p-groups of class 2 and exponent p in time polynomial in the group order. It is known that this problem can be reduced to solving the alternating matrix space isometry problem over a finite field in time polynomial in the underlying vector space size. We propose a venue of attack for the latter problem by viewing it as a linear algebraic analogue of the graph isomorphism problem. This viewpoint leads us to explore the possibility of transferring techniques for graph isomorphism to this long-believed bottleneck case of group isomorphism.**

In 1970's, Babai, Erdős, and Selkow presented the first average-case efficient graph isomorphism testing algorithm (SIAM J Computing, 1980). Inspired by that algorithm, we devise an average-case efficient algorithm for the alternating matrix space isometry problem over a key range of parameters, in a random model of alternating matrix spaces in vein of the Erdős-Rényi model of random graphs. For this, we develop a linear algebraic analogue of the classical individualisation technique, a technique belonging to a set of combinatorial techniques that has been critical for the progress on the worst-case time complexity for graph isomorphism, but was missing in the group isomorphism context. This algorithm also enables us to improve Higman's 57-year-old lower bound on the number of p-groups (Proc. of the LMS, 1960). We finally show that Luks' dynamic programming technique for graph isomorphism (STOC 1999) can be adapted to slightly improve the worst-case time complexity of the alternating matrix space isometry problem in a certain range of parameters.

Most notable progress on the worst-case time complexity of graph isomorphism, including Babai's recent breakthrough (STOC 2016) and Babai and Luks' previous record (STOC 1983), has relied on both group theoretic and combinatorial techniques. By developing a linear algebraic analogue of the individualisation technique and demonstrating its usefulness in the average-case setting, the main result opens up the possibility of adapting that strategy for graph isomorphism to this hard instance of group isomorphism. The linear algebraic Erdős-Rényi model is of independent interest and may deserve further study.

***Keywords*-group isomorphism; graph isomorphism; Erdős-Rényi model; individualisation and refinement;**

I. Introduction

A. Problems, postulates, and models

Let $\mathbb{F}_q$ be the finite field with q elements. An $n \times n$ matrix P over $\mathbb{F}_q$ is *alternating*, if for every $u \in \mathbb{F}_q^n$, $u^{\mathrm{t}} P u = 0$.[1] $\Lambda(n, q)$ denotes the linear space of $n \times n$ alternating matrices over $\mathbb{F}_q$, and a dimension-m subspace of $\Lambda(n, q)$ is called an m-alternating (matrix) space. $\mathrm{GL}(n, q)$ denotes the general linear group of degree n over $\mathbb{F}_q$. We study the following problem.

Problem 1 (Alternating matrix space isometry problem, AltMatSpIso)**.** Given the linear bases of two m-alternating spaces $\mathcal{G}, \mathcal{H}$ in $\Lambda(n, q)$, decide whether there exists $A \in \mathrm{GL}(n, q)$, such that $A^{\mathrm{t}}\mathcal{G}A := \{A^{\mathrm{t}} P A : P \in \mathcal{G}\}$ is equal to $\mathcal{H}$ as subspaces.

If such an A exists, we say that $\mathcal{G}$ and $\mathcal{H}$ are *isometric*. As will be explained in Section I-B, AltMatSpIso has been studied, mostly under other names, for decades. It lies at the heart of the group isomorphism problem (GroupIso), and has an intimate relationship with the celebrated graph isomorphism problem (GraphIso). As a problem in $\mathrm{NP} \cap \mathrm{coAM}$, its worst-case time complexity has barely been improved over the brute-force algorithm. In fact, to obtain $q^{O(n+m)}$-time algorithm is already regarded as very difficult.

Let us recall one formulation of GraphIso. For $n \in \mathbb{N}$, let $[n] = \{1, 2, \ldots, n\}$, and S_n denotes the symmetric group on $[n]$. A simple undirected graph is just a subset of $\Lambda_n := \{\{i, j\} : i, j \in [n], i \neq j\}$. A permutation $\sigma \in S_n$ induces a natural action on Λ_n. The following formulation of GraphIso as an instance of the setwise transporter problem is well-known [35].

Problem 2 (Graph isomorphism problem, GraphIso)**.** Given two subsets G, H of Λ_n, decide whether there exists

[1] A is skew-symmetric if $P^t = -P$. When $\mathbb{F}$ is of characteristic not 2, skew-symmetric and alternating are equivalent. When $\mathbb{F}$ is of characteristic 2, alternating implies skew-symmetric but not vice versa.

0272-5428/17 $31.00 © 2017 IEEE
DOI 10.1109/FOCS.2017.49

$\sigma \in S_n$, such that $G^\sigma := \{\{i^\sigma, j^\sigma\} : \{i, j\} \in G\}$ is equal to H as sets.

The formulations of ALTMATSPISO and GRAPHISO as in Problem 1 and Problem 2 lead us to the following postulate.

Postulate 1. ALTMATSPISO *can be viewed and studied as a linear algebraic analogue of* GRAPHISO.

Postulate 1 originates from the following meta-postulate.

Meta-postulate. *Alternating matrix spaces can be viewed and studied as a linear algebraic analogue of graphs.*

This meta-postulate will be studied further in [43]. As a related note, recent progress on the non-commutative rank problem suggests the usefulness of viewing linear spaces of matrices as a linear algebraic analogue of bipartite graphs [20, 25, 26].

From the meta-postulate, we formulate a model of random alternating matrix spaces over $\mathbb{F}_q$. Let $\left[\ \right]_q$ be the Gaussian binomial coefficient with base q.

Model 1 (The linear algebraic Erdős-Rényi model)**.** *The linear algebraic Erdős-Rényi model,* $\text{LINER}(n, m, q)$*, is the uniform probability distribution over the set of dimension-*m *subspaces of* $\Lambda(n, q)$*, that is, each subspace is endowed with probability* $1/\left[{n \atop m}\right]_q$.

Model 1 clearly mimics the usual Erdős-Rényi model.

Model 2 (Erdős-Rényi model)**.** *The Erdős-Rényi model* $\text{ER}(n, m)$ *is the uniform probability distribution over the set of size-*m *subsets of* Λ_n*, that is, each subset is endowed with probability* $1/\binom{n}{m}$.

We then pose the following postulate.

Postulate 2. $\text{LINER}(n, m, q)$ *can be viewed and studied as a linear algebraic analogue of* $\text{ER}(n, m)$.

B. Background of the alternating matrix space isometry problem

While the name ALTMATSPISO may be unfamiliar to some readers, this problem has been studied for decades as an instance – in fact, the long-believed bottleneck case – of the group isomorphism problem. This problem also has an intricate relationship with the graph isomorphism problem. We first review these connections below, and then examine the current status of this problem.

1) Relation with the group isomorphism problem: We first introduce the group isomorphism problem (GROUPISO) and mention a long-believed bottleneck instance of this problem. It turns out that ALTMATSPISO is almost equivalent to this instance.

GROUPISO asks to decide whether two finite groups of order n are isomorphic. The difficulty of this problem depends crucially on how we represent the groups in the algorithms. If our goal is to obtain an algorithm running in time $\text{poly}(n)$, then we may assume that we have at our disposal the Cayley (multiplication) table of the group, as we can recover the Cayley table from most reasonable models for computing with finite groups. Therefore, in the main text we restrict our discussion mostly to this very redundant model, which is meaningful mainly because we do not know a $\text{poly}(n)$-time or even an $n^{o(\log n)}$-time algorithm [50] ($\log$ to the base 2), despite that a simple $n^{\log n + O(1)}$-time algorithm has been known for decades [18, 40]. The past few years have witnessed a resurgence of activity on algorithms for this problem with worst-case analysis in terms of the group order; we refer the reader to [21] which contains a survey of these algorithms.

It is long believed that p-groups form the bottleneck case for GROUPISO. In fact, the decades-old quest for a polynomial-time algorithm has focused on class-2 p-groups, with little success. Even if we restrict further to consider p-groups of class 2 and exponent p, the problem is still difficult. Recent works [12, 13, 24, 31] solve some nontrivial subclasses of this group class, and have lead to substantial improvement in practical algorithms. But the methods in those works seem not helpful enough to lead to any improvement for the worst-case time complexity of the general class.

By a classical result of Baer [9], 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 $p^{O(m+n)}$. On the other hand, there also is an inverse reduction for $p > 2$. In fact, when such p-groups are given by generators in the permutation group quotient model [27], isomorphism testing reduces to solving ALTMATSPISO in time $\text{poly}(n, m, \log p)$ [12]. Because of these reductions and the current status of GROUPISO, we see that ALTMATSPISO lies at the heart of GROUPISO, and solving ALTMATSPISO in $q^{O(m+n)}$ is already very difficult.

We now recall the reductions mentioned in the last paragraph, which is classical by [9, 47], but seems not well-known among theoretical computer scientists. See also [49] and [21].

Suppose we are given two p-groups of class 2 and exponent p, G_1 and G_2 of order p^ℓ. For G_i, let $b_i : G_i/[G_i, G_i] \times G_i/[G_i, G_i] \to [G_i, G_i]$ be the commutator map where $[G_i, G_i]$ denotes the commutator subgroup. By the class 2 and exponent p assumption, $G_i/[G_i, G_i]$ are elementary abelian groups of exponent p. For G_1 and G_2 to be isomorphic it is necessary that $[G_1, G_1] \cong [G_2, G_2] \cong \mathbb{Z}_p^m$ and $G_1/[G_1, G_1] \cong G_2/[G_2, G_2] \cong \mathbb{Z}_p^n$ such that $m+n = \ell$. Furthermore b_i's are alternating bilinear maps. So we have alternating bilinear maps $b_i : \mathbb{F}_p^n \times \mathbb{F}_p^n \to \mathbb{F}_p^m$. G_1 and G_2 are isomorphic if and only if there exist $A \in \text{GL}(n, p)$ and $D \in \text{GL}(n, p)$ such that for every $u, v \in \mathbb{F}_p^n$, $b_1(A(u), A(v)) = D(b_2(u, v))$. The latter is also known as the *pseudo-isometry testing problem* of alternating maps. Representing b_1 and b_2 as tuples of alternating matrices $\mathbf{G}_1 = (P_1, \ldots, P_m) \in$

$\Lambda(n,p)^m$ and $\mathbf{G}_2 = (Q_1, \ldots, Q_m) \in \Lambda(n,p)^m$, it translates to ask whether $A^{\mathrm{t}}\mathbf{G}_1 A = \mathbf{G}_2^D$. Let $\mathcal{G}_1$ (resp. $\mathcal{G}_2$) be the linear span of $\mathbf{G}_1$ (resp. $\mathbf{G}_2$). This becomes an instance of ALTMATSPISO w.r.t. $\mathcal{G}_1$ and $\mathcal{G}_2$.

When $p > 2$, we can reduce ALTMATSPISO to isomorphism testing of p-groups of class 2 and exponent p using the following construction. Starting from $\mathbf{G} \in \Lambda(n,p)^m$ representing $\mathcal{G} \leq \Lambda(n,p)$, $\mathbf{G}$ can be viewed as representing a bilinar map $b : \mathbb{F}_p^n \times \mathbb{F}_p^n \to \mathbb{F}_p^m$. Define a group G with operation $\circ$ over the set $\mathbb{F}_p^m \times \mathbb{F}_p^n$ as $(v_1, u_1) \circ (v_2, u_2) = (v_1 + v_2 + \frac{1}{2}b(u_1, u_2), u_1 + u_2)$. It can be verified that G is a p-group of class 2 and exponent p, and it is known that two such groups G_1 and G_2 built from $\mathbf{G}_1$ and $\mathbf{G}_2$ are isomorphic if and only if $\mathcal{G}_1$ and $\mathcal{G}_2$ are isometric.

When working with groups in the Cayley table model, and working with ALTMATSPISO in time $p^{O(m+n)}$, the above reductions can be performed efficiently. In [12], it is discussed which models of computing with finite groups admit the reduction from isomorphism testing of p-groups of class 2 and exponent p to the pseudo-isometry testing of alternating bilinear maps. In particular, it is concluded there that the reduction works in the permutation group quotient model introduced in [27].

2) Relation with the graph isomorphism problem: The celebrated graph isomorphism problem (GRAPHISO) asks to decide whether two undirected simple graphs are isomorphic. The relation between ALTMATSPISO and GRAPHISO is very delicate. Roughly speaking, the two time-complexity measures of ALTMATSPISO, $q^{O(n+m)}$ and $\mathrm{poly}(n,m,q)$, sandwiches GRAPHISO in an interesting way. For one direction, solving ALTMATSPISO in time $q^{O(n+m)}$ can be reduced to solving GRAPHISO for graphs of size $q^{O(n+m)}$, by first reducing to solving GROUPISO for groups of order $q^{O(n+m)}$ as above, and then to solving GRAPHISO for graphs of size $q^{O(n+m)}$ by the reduction from GROUPISO to GRAPHISO [30]. Therefore, a polynomial-time algorithm for GRAPHISO implies an algorithm for ALTMATSPISO in time $q^{O(n+m)}$. It is then reasonable to examine whether the recent breakthrough of Babai [2, 3], a quasipolynomial-time algorithm for GRAPHISO, helps with reducing the time complexity of ALTMATSPISO. This seems unlikely. One indication is that the brute-force algorithm for ALTMATSPISO is already quasipolynomial with respect to $q^{O(n+m)}$. Another evidence is that Babai in [2, arXiv version 2, Section 13.2] noted that his algorithm seemed not helpful to improve GROUPISO, and posed GROUPISO as one roadblock for putting GRAPHISO in P. Since ALTMATSPISO captures the long-believed bottleneck case for GROUPISO, the current results for GRAPHISO are unlikely to improve the time complexity to $q^{O(n+m)}$. There is also an explanation from the technical viewpoint [19]. Roughly speaking, the barrier in the group theoretic framework for GRAPHISO is to deal with large alternating groups, as other composition factors like projective special linear groups can be handled by brute-force in quasipolynomial time, so for the purpose of a quasipolynomial-time algorithm these group are not a concern. On the other hand, for ALTMATSPISO, it is exactly the projective special linear groups that form a bottleneck. For the other direction, in a forthcoming work [22], it is shown that solving GRAPHISO in polynomial time reduces to solving ALTMATSPISO over $\mathbb{F}_q$ with $q = \mathrm{poly}(n)$ in time $\mathrm{poly}(n,m,q)$.

3) Current status of ALTMATSPISO*:* It is easy to see that solving ALTMATSPISO in $\mathrm{poly}(n,m,\log q)$ is in $\mathrm{NP} \cap \mathrm{coAM}$, so it is unlikely to be NP-complete. As to the worst-case time complexity, the brute-force algorithm for ALTMATSPISO runs in time $q^{n^2} \cdot \mathrm{poly}(n,m,\log q)$. Another analysed algorithm for ALTMATSPISO offers a running time of $q^{\frac{1}{4}(n+m)^2+O(n+m)}$ when $q = p > 2$ is a prime, by first reducing to testing isomorphism of class-2 and exponent-p p-groups of order p^{n+m}, and then applying Rosenbaum's $N^{\frac{1}{4}\log_p N + O(1)}$-time algorithm for p-groups of order N [44]. This is only better than the brute-force one when $m < n$.[2] It is somewhat embarrassing that for a problem in $\mathrm{NP} \cap \mathrm{coAM}$, we are only able to barely improve over the brute-force algorithm in a limited range of parameters. In a very true sense, our current understanding of the worst-case time complexity of ALTMATSPISO is like the situation for GRAPHISO in the 1970's.

On the other hand, practical algorithms for ALTMATSPISO have been implemented. As far as we know, current implemented algorithms for ALTMATSPISO can handle the case when $m + n \approx 20$ and $p \approx 13$, but absolutely not the case if $m + n \approx 200$, though for $m + n \approx 200$ and say $p \approx 13$ the input can be stored in a few megabytes.[3] For GRAPHISO, the programs NAUTY and TRACES [39] can test isomorphism of graphs stored in gigabytes in a reasonable amount of time. Therefore, unlike GRAPHISO, ALTMATSPISO seems hard even in the practical sense.

4) On the parameters: From the discussion above, we see that solving ALTMATSPISO with a worst-case time complexity $q^{O(n+m)}$ seems already a difficult target. From the meta-postulate, it is helpful to think of vectors in $\mathbb{F}_q^n$ as vertices, and matrices in an m-alternating space as edges, so the $q^{O(n+m)}$ measure can be thought of as polynomial in the number of "vertices" and the number of "edges." Here the parameter m comes into the theme, because q^m, while no more than $q^{\binom{n}{2}}$, is not necessarily bounded by a polynomial in q^n. This is in contrast to GRAPHISO, where the edge number is at most quadratic in the vertex

[2] As pointed out in [12], there are numerous unanalysed algorithms [16, 42] which may lead to some improvement, but $q^{cn^2} \cdot \mathrm{poly}(n,m,\log q)$ for some constant $0 < c < 1$ is a reasonable over estimate of the best bound by today's method.

[3] We thank James B. Wilson, who maintains a suite of algorithms for p-group isomorphism testing, for communicating his hands-on experience to us. We take the responsibility for any possible misunderstanding or not knowing of the performance of other implemented algorithms.

number. In particular, when $m = \Omega(n^2)$, then the brute-force algorithm which runs in $q^{n^2} \cdot \mathrm{poly}(m, n, \log q)$ is already in time $q^{O(n+m)}$. Furthermore, if we consider all $n \times n$ alternating matrix spaces (regardless of the dimension), most of them are of dimension $\Omega(n^2)$, so the brute-force algorithm already works in time $q^{O(n+m)}$ for most alternating matrix spaces. On the other hand, when m is very small compared to n, say $m = O(1)$, we can enumerate all elements in $\mathrm{GL}(m, q)$ in time $q^{O(1)}$, and apply the isometry testing for alternating matrix *tuples* from [24] which runs in randomised time $\mathrm{poly}(n, m, \log q)$. Therefore, the $q^{O(n+m)}$-time measure makes most sense when m is comparable with n, in particular when $m = \Theta(n)$. This is why we study average-case algorithms in this regime of parameters (e.g. LINER(n, m, q) with $m = \Theta(n)$), while the average-case algorithm for GRAPHISO in [5] considers all graphs (e.g. each labelled graph is taken with probability $1/2^{\binom{n}{2}}$).

C. Algorithmic results

Postulates 1 and 2 seem hopeful at first sight by the formulations of ALTMATSPISO and LINER. But realities in the combinatorial world and the linear algebraic world can be quite different, as just discussed in the last paragraph. So meaningful results cannot be obtained by adapting the results for graphs to alternating matrix spaces in a straightforward fashion. One purpose of this article is to provide evidence that, despite potential technical difficulties, certain ideas that have been developed for GRAPHISO and ER can be adapted to work with ALTMATSPISO and LINER.

We will take a shortcut, by presenting one result that supports both postulates. In the graph setting, such a result is naturally an efficient graph isomorphism testing algorithm with an average-case analysis in the Erdős-Rényi model. The first such algorithm was proposed by Babai, Erdős and Selkow in 1970's [5], with follow-up improvements by Lipton [34], Karp [28], and Babai and Kučera [6]. Therefore we set to study average-case algorithms for ALTMATSPISO in the LINER model. Inspired by the algorithm in [5], we show the following.

Theorem 1 (Main result)**.** *Suppose $m = cn$ for some constant c. There is an algorithm which, for all but at most $1/q^{\Omega(n)}$ fraction of alternating matrix spaces $\mathcal{G}$ in* LINER(n, m, q)*, tests any alternating matrix space $\mathcal{H}$ for isometry to $\mathcal{G}$ in time $q^{O(n)}$.*

An important ingredient in Theorem 1, the utility of which should go beyond the average-case setting, is an adaptation of the *individualisation* technique for GRAPHISO to ALTMATSPISO. We also realise a reformulation of the *refinement* technique for GRAPHISO as used in [5] in the ALTMATSPISO setting. Individualisation and refinement are very influential combinatorial ideas for GRAPHISO, have been crucial in the progress of the worst-case time complexity of GRAPHISO, including Babai's recent breakthrough [2, 3], but were missing in the GROUPISO context.

> *The main contribution of this article to* ALTMATSPISO *is to initiate the use of the individualisation and refinement ideas for* GRAPHISO *in this problem.*

Here, we note an interesting historical coincidence. Babai was the first to import the group theoretic idea to GRAPHISO in 1979 [1], by when the combinatorial techniques had been around for quite some time. On the other hand, we have an opposite situation for ALTMATSPISO: the relevant group theoretic tools have been the subject of intensive study for decades, while it is the combinatorial individualisation and refinement ideas that need to be imported. We do understand though, that there are valid reasons for people not having come to this before. For example, we would not have come to such ideas, if we restrict ourselves to solving ALTMATSPISO in time $\mathrm{poly}(n, m, \log q)$. In Section IV-A, we will reflect on the historical development on the worst-case complexity of GRAPHISO, and discuss the prospect of getting a $q^{O(n^{2-\epsilon})}$-time algorithm for ALTMATSPISO.

For an m-alternating space $\mathcal{G}$ in $\Lambda(n, q)$, define the autometry group of $\mathcal{G}$, $\mathrm{Aut}(\mathcal{G})$ as $\{A \in \mathrm{GL}(n, q) : A^t \mathcal{G} A = \mathcal{G}\}$. The proof of Theorem 1 implies the following, which can be viewed as a weaker correspondence of the classical result that most graphs have trivial automorphism groups [17].

Corollary 2. *Suppose $m = cn$ for some constant c. All but $1/q^{\Omega(n)}$ fraction of alternating matrix spaces in* LINER(n, m, q) *have autometry groups of size $q^{O(n)}$.*

Finally, we provide another piece of evidence to support Postulate 1, by adapting Luks' dynamic programming technique for GRAPHISO [38] to ALTMATSPISO. In the GRAPHISO setting, this technique improves the naive $n! \cdot \mathrm{poly}(n)$ time bound to the $2^{O(n)}$ time bound, which can be understood as replacing the number of permutations $n!$ with the number of subsets 2^n. In the linear algebraic setting the analogue would be to replace $\Theta(q^{n^2})$, the number of invertible matrices over $\mathbb{F}_q$, with the number of subspaces in $\mathbb{F}_q^n$ which is $q^{\frac{1}{4}n^2 + O(n)}$. We show that this is indeed possible.

Theorem 3. *There exists a deterministic algorithm for* ALTMATSPISO *in time $q^{\frac{1}{4}(m^2+n^2)+O(m+n)}$.*

Note that the quadratic term on the exponent of the algorithm in Theorem 3 is $\frac{1}{4}(m^2 + n^2)$, slightly better than the one based on Rosenbaum's result [44] which is $\frac{1}{4}(m + n)^2$. We stress though that our intention to present this result is to support Postulate 1.

D. Applications to enumeration of finite p-groups

A basic question in finite group theory is to determine the number of groups of a given order n (up to isomorphism). When n is a prime power, this was first studied by G. Higman in 1960 [23], who showed that the number of p-groups of order p^ℓ is lower bounded by $p^{\frac{2}{27}\ell^3 - \frac{4}{9}\ell^2}$, by

consider p-groups of Frattini class 2. (A p-group is of Frattini class 2 if there exists a central elementary abelian group such that its quotient is also elementary abelian.) Sims established an upper bound $p^{\frac{2}{27}\ell^3+O(\ell^{8/3})}$ [46], later improved to $p^{\frac{2}{27}\ell^3+O(\ell^{5/2})}$ by Newman and Seely (see [11, Sec. 5]). We refer the interested reader to the excellent monograph [11] for more on this research direction. By the correspondence between p-groups of class 2 and exponent p and alternating matrix spaces, Corollary 2, with appropriate choices of parameters, implies that the number of p-groups of class 2 and exponent p is lower bounded by $p^{\frac{2}{27}\ell^3-\frac{2}{9}\ell^2-O(\ell)}$. This already improves Higman's 57-year-old lower bound. Note that our lower bound is established by considering p-groups of class 2 and exponent p, which is a subclass of p-groups of Frattini class 2.

Theorem 4. *The number of p-groups of order p^ℓ is lower bounded by $p^{\frac{2}{27}\ell^3-\frac{2}{9}\ell^2-O(\ell)}$.*

On the other hand, the number of p-groups of class 2 and exponent p of order ℓ is upper bounded by $\ell \cdot p^{\frac{2}{27}\ell^3-\frac{2}{9}\ell^2+\frac{49}{72}\ell}$ [11, Theorem 19.3]. Our Theorem 4 then implies that the coefficient of the quadratic term on the exponent is $-2/9$ for such p-groups. This answers an open problem in [11], namely Question 22.8 in the case of this group class. The proof of Theorem 4 can be viewed as adapting the techniques from random graph theory to study p-groups. This suggests that the linear algebraic Erdős-Rényi model may deserve further study. We will make some general remarks on this in Section IV-B.

In a follow-up work [32], we will extend our techniques in two ways to yield stronger results for enumerating p-groups. The first improvement will be to get the exact coefficient of the linear term on the exponent. The second improvement will be to deal with p-groups of Frattini class 2.

Organisation of this extended abstract. In Section II, we present the outline of the algorithm for Theorem 1. In Section III, we demonstrate the dynamic programming idea using the subspace transporter problem, and present an outline of the proof of Theorem 3. Section IV includes discussions, future directions, and a review of the relation between p-group isomorphism testing and alternating matrix space isometry testing. Due to lack of space, some proofs have to be omitted from this extended abstract. Details can be found in the full version of this paper [33].

II. Outline of the main algorithm

We now describe the outline of the algorithm for Theorem 1, which is inspired by the first average-case efficient algorithm for GRAPHISO by Babai, Erdős, and Selkow [5]. We will recall the idea in [5] that is relevant to us, define a linear algebraic individualisation, and propose a reformulation of the refinement step in [5]. Then we present an outline of the main algorithm. During the procedure we will also see how the meta-postulate guides the generalisation here.

A. A variant of the naive refinement algorithm as used in [5]

Two properties of random graphs are used in the average-case analysis of the algorithm in [5]. The first property is that most graphs have the first $\lceil 3\log n \rceil$ largest degrees distinct. The second property, which is relevant to us, is the following.

Let $G = ([n], E)$ be a simple undirected graph (the labeling of vertices are determined corresponding to their degrees in the decreasing order). Let $r = \lceil 3\log n \rceil$, and $S = [r]$, $T = [n] \setminus [r]$. Let B be the bipartite graph induced by the cut $[r] \cup \{r+1, \ldots, n\}$, that is, $B = (S \cup T, F)$ where $F = \{(i,j) : i \in S, j \in T, \{i,j\} \in E\}$. For each $j \in T$, assign a length-r bit string f_j as follows: $f_j \in \{0,1\}^r$ such that $f_j(i) = 1$ if and only if $(i,j) \in F$. It is easy to verify that, all but at most $O(1/n)$ fraction of graphs satisfy that f_j's are distinct over $j \in T$.

Let us see how the second property alone, together with the individualisation and refinement heuristic, give an average-case algorithm in $n^{O(\log n)}$. Suppose G satisfies the second property, and we would like to test isomorphism between $G = ([n], E)$ and an arbitrary graph $H = ([n], E')$. Let $St_G \subseteq \{0,1\}^r$ be the set of bit strings obtained in the procedure above. Note that $|St_G| = n - r$. In the *individualising* step, we enumerate all r-tuple of vertices in H. For a fixed r-tuple $(i_1, \ldots, i_r) \in [n]^r$, we perform the *refinement* step, that is, label the remaining vertices in H according to their adjacency relations with the r-tuple $(i_1, \ldots, i_r)$ as before, to obtain another set of bit-strings St_H. If $St_G \neq St_H$ we neglect this r-tuple. If $St_G = St_H$, then form a bijective map between $[n]$ and $[n]$, by mapping j to i_j for $j \in [r]$, and the rest according to their labels. Finally check whether this bijective map induces an isomorphism.

It can be verified easily that the above algorithm is an $n^{O(\log n)}$-time algorithm that tests isomorphism between G and H given that G satisfies the required property. In particular, this implies that for such G, $|\mathrm{Aut}(G)| \leq n^{O(\log n)}$. To recover the algorithm in [5], assuming that the largest r degrees are distinct, one can canonicalise the choice of the r-tuples by choosing the one with largest r degrees for both G and H.

B. Individualisation and refinement in the ALTMATSPISO setting

We will generalise the above idea to the setting of ALTMATSPISO. To do this, we first make sense of what individualisation means in the alternating space setting. We discuss how the refinement step may be generalised, and indicate how we follow an alternative formulation of it.

Let $G = ([n], E)$ and $H = ([n], E')$ be two graphs for which we want to test isomorphism. Let $\mathcal{G}, \mathcal{H} \leq \Lambda(n, q)$ be two m-alternating spaces for which we want to test isometry. As the case in Section II-A, we will look for properties of

G or $\mathcal{G}$ which enable the average-case analysis, and perform individualisation on H or $\mathcal{H}$ side.

For $i \in [n]$, e_i denotes the ith standard basis vector of $\mathbb{F}_q^n$. For a vector space V and $S \subseteq V$, we use $\langle S \rangle$ to denote the linear span of S in V.

Individualisation. In the graph setting, individualising r vertices in H can be understood as follows. First we fix a size-r subset L of $[n]$. Then put an order on the elements in L. The result is a tuple of distinct vertices $(i_1, \ldots, i_r) \in [n]^r$. Enumerating such tuples incurs a multiplicative cost of at most n^r.

In the alternating matrix space setting, it is helpful to think of vectors in $\mathbb{F}_q^n$ as vertices, and matrices in $\mathcal{H}$ as edges. Consider the following procedure. First fix a dimension-r subspace L of $\mathbb{F}_q^n$. Then choose an ordered basis of L. The result is a tuple of linearly independent vectors $(v_1, \ldots, v_r)$, $v_i \in \mathbb{F}_q^n$, such that $L = \langle v_1, \ldots, v_r \rangle$. This incurs a multiplicative cost of at most q^{rn}. Up to this point, this is in complete analogy with the graph setting. We may stop here and say that an r-individualisation amounts to fix an r-tuple of linearly independent vectors.

We can go a bit further though. As will be clear in the following, it is beneficial if we also fix a complement subspace R of L, i.e. $R \leq \mathbb{F}_q^n$ such that $L \cap R = \{0\}$ and $\langle L \cup R \rangle = \mathbb{F}_q^n$. This adds another multiplicative cost of $q^{r(n-r)}$, which is the number of complement subspaces of a fixed dimension-r subspace in $\mathbb{F}_q^n$. In the graph setting, this step is not necessary, because for any $L \subseteq [n]$ there exists a unique complement subset $R = [n] \setminus L$.

To summarise, by an *r-individualisation*, we mean choosing a direct sum decomposition $\mathbb{F}_q^n = L \oplus R$ where $\dim(L) = r$ and $\dim(R) = n - r$, together with an ordered basis $(v_1, \ldots, v_r)$ of L. Enumerating all r-individualisations incurs a total multiplicative cost of at most q^{2rn-r^2}.

Towards a refinement step as in [5]. In the GRAPHISO setting, individualising r vertices gives $(i_1, \ldots, i_r) \in [n]^r$, and allows us to focus on isomorphisms that respect this individualisation, namely those $\phi \in \mathrm{Iso}(G, H)$ such that $\phi(j) = i_j$ for $j \in [r]$. There are at most $(n-r)!$ such isomorphisms. Since r is usually set as a polylog, just naively trying all such permutations does not help. Therefore the individualisation is usually accompanied with a refinement type technique.

Specifically, setting $L = \{i_1, \ldots, i_r\}$ and $R = [n] \setminus L$, the refinement step as in [5] assigns every $v \in R$ a label according to its adjacency relation w.r.t. $(i_1, \ldots, i_r)$. This label in fact represents a *subset* of L, and an individualisation-respecting isomorphism has to preserve this adjacency relation for every $v \in R$. This restriction turns out to be quite severe for most graphs: as observed in Section II-A, for most graphs G, the adjacency relations between $(1, 2, \ldots, r)$ and $j \in [n] \setminus [r]$ are completely different over j. For such G and any individualisation of H, this means that there is at most one way to extend $\phi(j) = i_j$ for $j \in [r]$ to an isomorphism between G and H.

In the ALTMATSPISO setting, an r-individualisation also allows us to focus on isometries that respect the decomposition $L \oplus R$ and the ordered basis $(v_1, \ldots, v_r)$ of L, namely those $\phi \in \mathrm{Iso}(\mathcal{G}, \mathcal{H})$ such that $\phi(e_i) = v_i$ for $i \in [r]$, and $\phi(\langle e_{r+1}, \ldots, e_n \rangle) = R$. There are at most $q^{(n-r)^2}$ such isometries. Since r will be also set to be very small – in fact a constant here – we also need some refinement type argument. For $u \in R$, we can record its "adjacency relation" w.r.t. $\mathbf{v} = (v_1, \ldots, v_r)$ as a *subspace* of $L \cong \mathbb{F}_q^r$ as follows. For $Q \in \mathcal{H} \leq \Lambda(n, q)$, define $Q(\mathbf{v}, u) := (v_1^{\mathrm{t}} Q u, \ldots, v_r^{\mathrm{t}} Q u)^{\mathrm{t}} \in \mathbb{F}_q^r$, and $\mathcal{H}(\mathbf{v}, u) := \{Q(\mathbf{v}, u) : Q \in \mathcal{H}\}$ which is a subspace in $\mathbb{F}_q^r$. $\mathcal{H}(\mathbf{v}, u)$ records the adjacency relation between $(v_1, \ldots, v_r)$ and u under $\mathcal{H}$. It can be verified that an individualisation-respecting isometry has to preserve this adjacency relation. It is tempting to check then on the $\mathcal{G}$ side, where we have the standard individualisation $(e_1, \ldots, e_r)$ and $\langle e_{r+1}, \ldots, e_n \rangle$, whether for most $\mathcal{G}$'s it is the case that every $v \in \langle e_{r+1}, \ldots, e_n \rangle$ gets a unique label. If this is so, then the number of individualisation-respecting isomorphisms can also be significantly reduced. However, this cannot be the case when r is small, as there are $q^{(n-r)^2}$ vectors in R but there at at most q^{r^2} subspaces in $\mathbb{F}_q^r$.

The alert reader will note that, since we are looking for linear maps from $\langle e_{r+1}, \ldots, e_n \rangle$ to R, the above counting argument does not make much sense, as it mostly concerns setwise maps from $\langle e_{r+1}, \ldots, e_n \rangle$ to R. It is indeed the case, and we further note that the map from $u \in R$ to $\mathcal{H}(\mathbf{v}, u) \leq \mathbb{F}_q^r$ defines a sheaf over the projective space $\mathbb{P}(R)$, so such labels have some nontrivial relation to glue together to form a sheaf. (See the related concept of kernel sheaves as in [29].) It may be possible to use these observations to define a reasonable refinement step in the alternating matrix space setting. In this paper we shall follow the following reformulation.

A reformulation of the refinement step. To resolve the above problem, we reformulate the idea in the graph setting as follows. Recall that on the G side we start with the standard individualisation $[r] \cup \{r+1, \ldots, n\}$ with an order on $[r]$ as $(1, \ldots, r)$, and let $S = [r]$, $T = \{r+1, \ldots, n\}$. This defines the bipartite graph $B = (S \cup T, F)$ where the edge set F is induced from G. For a fixed individualisation on the H side, which produces $L \cup R$, $L = \{i_1, \ldots, i_r\} \subseteq [n]$ with an order on L, this also defines a bipartite graph $C = (L \cup R, F')$ where F' is induced from H. A bijective $\psi : T \to R$ is a right-side isomorphism between B and C if it induces an isomorphism between B and C as bipartite graphs. Let $\mathrm{RIso}(B, C)$ be the set of right-side isomorphisms, and let $\mathrm{IndIso}(G, H)$ be the set of individualisation-respecting isomorphisms from G to H w.r.t the above individualisations. Note that both $\mathrm{RIso}(B, C)$ and $\mathrm{IndIso}(G, H)$ can be embedded to the set of bijective maps between T

and R. The key observation is that an individualisation-respecting isomorphism has to be a right-side isomorphism between B and C, e.g. $\mathrm{IndIso}(G,H) \subseteq \mathrm{RIso}(B,C)$. Also note that either $|\mathrm{RIso}(B,C)| = 0$ (e.g. when B and C are not right-isomorphic), or $|\mathrm{RIso}(B,C)| = |\mathrm{RAut}(B)|$ where $\mathrm{RAut}(B) := \mathrm{RIso}(B,B)$. The refinement step as in Section II-A achieves two goals. Firstly on the G side, most G's have the corresponding B with $|\mathrm{RAut}(B)| = 1$. This means that $|\mathrm{RIso}(B,C)| \leq 1$. Secondly, given H with a fixed individualisation inducing the corresponding bipartite graph C, there is an efficient procedure to decide whether B and C are right-isomorphic (by comparing the labels), and if they do, enumerate all right-isomorphisms (actually unique).

In the ALTMATSPISO setting, on the $\mathcal{G}$ side we start with the standard individualisation $S = \langle e_1, \ldots, e_r \rangle$, $T = \langle e_{r+1}, \ldots, e_n \rangle$ with the ordered basis $(e_1, \ldots, e_r)$ of S. We can also define a correspondence of the bipartite graph B in this setting, which is the matrix space $\mathcal{B}' = \{[e_1, \ldots, e_r]^t P [e_{r+1}, \ldots, e_n] : P \in \mathcal{G}\} \leq M(r \times (n-r), q)$, where $[e_1, \ldots, e_r]$ denotes the $n \times r$ matrix listing the column vectors $\{e_i : i = 1, \ldots, r\}$. Note that $[e_1, \ldots, e_r]^t P [e_{r+1}, \ldots, e_n]$ is just the upper-right $r \times (n-r)$ submatrix of P. Similarly, the individualisation on the $\mathcal{H}$ side yields $L \oplus R$ with an ordered basis of L, $(v_1, \ldots, v_r)$, $v_i \in \mathbb{F}_q^n$. Take any basis of $R = \langle v_{r+1}, \ldots, v_n \rangle$. Similarly construct $\mathcal{C}' = \{[v_1, \ldots, v_r]^t Q [v_{r+1}, \ldots, v_n] : Q \in \mathcal{H}\} \leq M(r \times (n-r), q)$. $A \in \mathrm{GL}(n-r, q)$ is a right-side equivalence between $\mathcal{B}'$ and $\mathcal{C}'$ if $\mathcal{B}'A := \{B'A : B' \in \mathcal{B}'\} = \mathcal{C}'$. Let $\mathrm{RIso}(\mathcal{B}', \mathcal{C}')$ be the set of right-side equivalences between $\mathcal{B}'$ and $\mathcal{C}'$, and $\mathrm{IndIso}(\mathcal{G}, \mathcal{H})$ the set of individualisation-respecting isometries between $\mathcal{G}$ and $\mathcal{H}$. Similarly, both $\mathrm{RIso}(\mathcal{B}', \mathcal{C}')$ and $\mathrm{IndIso}(\mathcal{G}, \mathcal{H})$ can be embedded in the set of invertible linear maps from T to R (isomorphic to $\mathrm{GL}(n-r, q)$), and we have $\mathrm{IndIso}(\mathcal{G}, \mathcal{H}) \subseteq \mathrm{RIso}(\mathcal{B}', \mathcal{C}')$. Furthermore $\mathrm{RIso}(\mathcal{B}', \mathcal{C}')$ is either empty (e.g. $\mathcal{B}'$ and $\mathcal{C}'$ are not right-side equivalent), or a coset of $\mathrm{RAut}(\mathcal{B}') := \mathrm{RIso}(\mathcal{B}', \mathcal{B}')$. So in analogy with the graph setting, for our purpose the goals become: (1) for most m-alternating space $\mathcal{G} \leq \Lambda(n, q)$ with $m = cn$ for some constant c, setting r to be some constant, we have $|\mathrm{RAut}(\mathcal{B}')| \leq q^{O(n)}$, and (2) for $\mathcal{G}$'s satisfying (1), $\mathrm{RIso}(\mathcal{B}', \mathcal{C}')$ can be enumerated efficiently.

We are almost ready for the algorithm outline. Alas, there is still one important ingredient missing. It turns out for the purpose of (2), we will need to "linearise" $\mathrm{RAut}(\mathcal{B}')$ to allow for the use of efficient linear algebra procedures. This linearisation is captured by the adjoint algebra concept, defined below in the algorithm outline. Correspondingly, in the goals above we will replace $\mathrm{RAut}(\mathcal{B}')$ and $\mathrm{RIso}(\mathcal{B}', \mathcal{C}')$ with $\mathrm{Adj}(\mathbf{B})$ and $\mathrm{Adj}(\mathbf{B}, \mathbf{C})$ where $\mathbf{B}$ and $\mathbf{C}$ will be defined below as well.

C. Algorithm outline

Suppose we want to test isometry between two m-alternating spaces $\mathcal{G} = \langle P_1, \ldots, P_m \rangle$ and $\mathcal{H} = \langle Q_1, \ldots, Q_m \rangle$ in $\Lambda(n, q)$. To ease the presentation in this subsection we assume $r = 4$ and $m = n - 4$.

We first define the property on $\mathcal{G}$ for the sake of average-case analysis. Given those $P_k \in \Lambda(n, q)$ linearly spanning $\mathcal{G}$, form a 3-tensor $\mathbf{G} \in \mathbb{F}_q^{n \times n \times m}$ where $\mathbf{G}(i, j, k)$ denotes the (i, j)th entry of P_k. Let $\mathbf{B}'$ be the upper-right $r \times (n-r) \times m$ subtensor of $\mathbf{G}$, with B'_k being the corresponding corner in P_k. B'_k's span the $\mathcal{B}'$ as defined above, so $A \in \mathrm{RAut}(\mathcal{B}') \leq \mathrm{GL}(n-r, q)$ if and only if there exists $D = (d_{i,j})_{i,j \in [m]} \in \mathrm{GL}(m, q)$ such that $\forall i \in [m]$, $\sum_{j \in [m]} d_{i,j} B'_j = B'_i A$. It is more convenient that we flip $\mathbf{B}'$ which is of size $r \times (n-r) \times m$ to the $\mathbf{B}$ which is of size $(n-r) \times m \times r$ (Figure 1). Slicing $\mathbf{B}$ along the third index, we obtain an r-tuple of $(n-r) \times m$ matrices $(B_1, \ldots, B_r)$ (Figure 2).

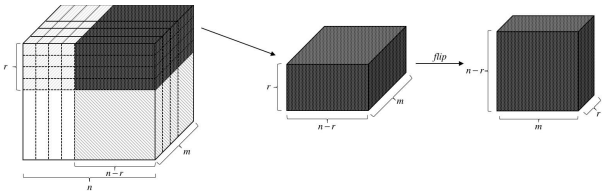

Figure 1. The 3-tensor $\mathbf{G}$, and flipping $\mathbf{B}'$ to get $\mathbf{B}$.

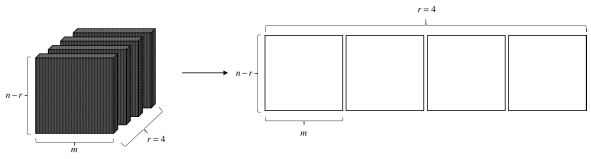

Figure 2. Slicing $\mathbf{B}$.

Define the set of equivalences of $\mathbf{B}$ as $\mathrm{Aut}(\mathbf{B}) := \{(A, D) \in \mathrm{GL}(n-r, q) \times \mathrm{GL}(m, q) : \forall i \in [r], AB_iD^{-1} = B_i\}$. Note that $\mathrm{RAut}(\mathcal{B}')$ is the projection of $\mathrm{Aut}(\mathbf{B})$ to the first component. Now define the adjoint algebra of $\mathbf{B}$ as $\mathrm{Adj}(\mathbf{B}) := \{(A, D) \in M(n-r, q) \oplus M(m, q) : \forall i \in [r], AB_i = B_iD\}$. $(A, D) \in M(n-r, q) \oplus M(m, q)$ is called invertible, if both A and D are invertible. Clearly, $\mathrm{Aut}(\mathbf{B})$ consists of the invertible elements in $\mathrm{Adj}(\mathbf{B})$. When $r = 4$, $m = n - r = n - 4$, it can be shown that the adjoint algebra of 4 random matrices in $M(m, q)$ is of size $q^{O(m)}$ with probability $1 - 1/q^{\Omega(m)}$. The key to prove this statement is the stable notion from geometric invariant theory [41] in the context of the left-right action of $\mathrm{GL}(m, q) \times \mathrm{GL}(m, q)$ on matrix tuples $M(m, q)^r$. In this context, a matrix tuple $(B_1, \ldots, B_r) \in M(m, q)^r$ is stable, if for every nontrivial subspace $U \leq \mathbb{F}_q^n$, $\dim(\langle \cup_{i \in [r]} B_i(U) \rangle) > \dim(U)$. An upper bound on $|\mathrm{Adj}(\mathbf{B})|$ can be obtained by analysing this notion using some classical algebraic results and elementary probability calculations. The good property we impose on

$\mathcal{G}$ is then that the corresponding $|\mathrm{Adj}(\mathbf{B})| \leq q^{O(m)}$. It can be verified that this property does not depend on the choices of bases of $\mathcal{G}$. There is one subtle point though: the analysis on $\mathrm{Adj}(\mathbf{B})$ is done for 4 random matrices but we want an analysis for $\mathcal{G}$ in the linear algebraic Erdős-Rényi model. This can be fixed by defining a so-called naive model and analysing the relation between the naive model and the LINER model. Roughly speaking, the naive model is a probability distribution on m-tuples of $n \times n$ alternating matrices, where each upper triangular entry of each alternating matrix is chosen from $\mathbb{F}_q$ in uniform random independently.

We have achieved our first goal, namely defining a good property satisfied by most $\mathcal{G}$'s. Let us see how this property enables an algorithm for such $\mathcal{G}$'s. For an arbitrary $\mathcal{H} \leq \Lambda(n,q)$, at a multiplicative cost of $q^{O(n)}$ (recall that $r = 4$) we can enumerate all r-individualisations. Consider a fixed one, say $\mathbb{F}_q^n = L \oplus R$ with an ordered basis $(v_1, \ldots, v_r)$ of L. Analogous to the above, we can construct $\mathbf{C}'$, flip to get $\mathbf{C}$, and slice $\mathbf{C}$ into r $(n-r) \times m$ The task then becomes to compute $\mathrm{Adj}(\mathbf{B}, \mathbf{C}) := \{(A, D) \in M(n-r, q) \oplus M(m, q) : \forall i \in [r], AB_i = C_i D\}$. Viewing A and D as variable matrices, $AB_i = C_i D$ are linear equations on A and D, so the solution set can be computed efficiently. As $|\mathrm{Adj}(\mathbf{B})| \leq q^{O(m)}$, for $\mathrm{Adj}(\mathbf{B}, \mathbf{C})$ to contain an invertible element, it must be that $|\mathrm{Adj}(\mathbf{B}, \mathbf{C})| = |\mathrm{Adj}(\mathbf{B})| \leq q^{O(m)}$. In this case all elements in $\mathrm{Adj}(\mathbf{B}, \mathbf{C})$ can be enumerated in time $q^{O(m)} = q^{O(n)}$. For each element $(A, D) \in \mathrm{Adj}(\mathbf{B}, \mathbf{C})$, test whether it is invertible, and if so, test whether the A in that solution induces an isometry together with the individualisation. This completes a high-level description of the algorithm. In particular, this implies that if $\mathcal{G}$ satisfies this property, then $|\mathrm{Aut}(\mathcal{G})| \leq q^{O(n)}$. A detailed presentation is in the full version [33], which have some minor differences with the outline here, as we want to reduce some technical details.

III. DYNAMIC PROGRAMMING

In this section, given a matrix group $G \leq \mathrm{GL}(n, q)$, we view G as a permutation group on the domain $\mathbb{F}_q^n$, so basic tasks like membership testing and pointwise transporter can be solved in time $q^{O(n)}$ by permutation group algorithms. Furthermore a generating set of G of size $q^{O(n)}$ can also be obtained in time $q^{O(n)}$. These algorithms are classical and can be found in [36, 45].

As mentioned in Section I, for GRAPHISO, Luks' dynamic programming technique [38] can improve the brute-force $n! \cdot \mathrm{poly}(n)$ time bound to the $2^{O(n)}$ time bound, which can be understood as replacing the number of permutations $n!$ with the number of subsets 2^n.

In our view, Luks' dynamic programming technique is most transparent when working with the subset transporter problem. Given a permutation group $P \leq S_n$ and $S, T \subseteq [n]$ of size k, this technique gives a $2^k \cdot \mathrm{poly}(n)$-time algorithm to compute $P_{S \to T} := \{\sigma \in P : \sigma(S) = T\}$ [8]. To illustrate the idea in the matrix group setting, we start with the subspace transporter problem.

Problem 3 (Subspace transporter problem)**.** Let $G \leq \mathrm{GL}(n, q)$ be given by a set of generators, and let V, W be two subspaces of $\mathbb{F}_q^n$ of dimension k. The subspace transporter problem asks to compute the coset $G_{V \to W} = \{g \in G : g(V) = W\}$.

The subspace transporter problem admits the following brute-force algorithm. Fix a basis $(v_1, \ldots, v_k)$ of V, and enumerate all ordered basis of W at the multiplicative cost of q^{k^2}. For each ordered basis $(w_1, \ldots, w_k)$ of W, compute the coset $\{g \in G : \forall i \in [k], g(v_i) = w_i\}$ by using a sequence of pointwise stabiliser algorithms. This gives an algorithm running in time $q^{k^2 + O(n)}$. Analogous to the permutation group setting, we aim to replace $O(q^{k^2})$, the number of ordered basis of $\mathbb{F}_q^k$, with $q^{\frac{1}{4}k^2 + O(k)}$, the number of subspaces in $\mathbb{F}_q^k$, via a dynamic programming technique. For this we first observe the following.

Observation 5. *There exists a deterministic algorithm that enumerates all subspaces of $\mathbb{F}_q^n$, and for each subspace computes an ordered basis, in time $q^{\frac{1}{4}n^2 + O(n)}$.*

Proof: For $d \in \{0, 1, \ldots, n\}$, let S_d be the number of dimension-d subspaces of $\mathbb{F}_q^n$. The total number of subspaces in $\mathbb{F}_q^n$ is $S_0 + S_1 + \cdots + S_n = q^{\frac{1}{4}n^2 + O(n)}$. To enumerate all subspaces we proceed by induction on the dimension in an increasing order. The case $d = 0$ is trivial. For $d \geq 1$, suppose all subspaces of dimension $d-1$, each with an ordered basis, are listed. To list all subspaces of dimension d, for each dimension-$(d-1)$ subspace U' with an ordered basis $(u_1, \ldots, u_{d-1})$, for each vector $u_d \notin U'$, form U with the ordered basis $(u_1, \ldots, u_d)$. Then test whether U has been listed. If so discard it, and if not add U together with this ordered basis to the list. The two for loops as above adds a multiplicative factor of at most $S_{d-1} \cdot q^n$, and other steps are basic linear algebra tasks. Therefore the total complexity is $\sum_{i=0}^{n} S_i \cdot q^{O(n)} = q^{\frac{1}{4}n^2 + O(n)}$. ∎

Theorem 6. *There exists a deterministic algorithm that solves the subspace transporter problem in time $q^{\frac{1}{4}k^2 + O(n)}$.*

Proof: We fix an ordered basis $(v_1, \ldots, v_k)$ of V, and for $d \in [k]$, let $V_d = \langle v_1, \ldots, v_d \rangle$. The dynamic programming table is a list, indexed by subspaces $U \leq W$. For $U \leq W$ of dimension $d \in [k]$, the corresponding cell will store the coset $G(V_d \to U) = \{g \in G : g(V_d) = U\}$. When $d = k$ the corresponding cell gives $G(V \to W)$.

We fill in the dynamic programming table according to d in an increasing order. For $d = 0$ the problem is trivial. Now assume that for some $d \geq 1$, we have computed $G(V_l \to U')$ for all $0 \leq l \leq d-1$ and subspace $U' \leq W$ of dimension U. To compute $G(V_d \to U)$ for some fixed $U \leq W$ of dimension d, note that any $g \in G(V_d \to U)$ has to map V_{d-1} to some $(d-1)$-dimension subspace $U' \leq U$, and v_d

to some vector $u \in U \setminus U_0$. This shows that

$$G(V_d \to U) = \bigcup_{U' \le U, \dim(U')=d-1} \bigcup_{u \in U \setminus U'} [G(V_{d-1} \to U')](v_d \to u).$$

To compute $[G(V_{d-1} \to U')](v_d \to u)$, we read $G(V_{d-1} \to U')$ from the table, then compute $[G(V_{d-1} \to U')](v_d \to u)$ using the pointwise transporter algorithm. The number of u in $U \setminus U'$ is no more than q^d, and the number of $(d-1)$-dimension subspaces of U is also no more than q^d. After taking these two unions, apply Sims' method to get a generating set of size $q^{O(n)}$. Therefore for each cell the time complexity is $q^{2d} \cdot q^{O(n)} = q^{O(n)}$. Therefore the whole dynamic programming table can be filled in time $q^{\frac{1}{4}k^2+O(k)} \cdot q^{O(n)} = q^{\frac{1}{4}k^2+O(n)}$. ∎

The proof of Theorem 6 contains the essential idea of how to use dynamic programming in this setting. We will apply this idea in a more sophisticated way to prove Theorem 3. The first step is to deal with the following problem.

Problem 4 (Alternating matrix transporter problem)**.** Let $H \le \mathrm{GL}(n,q)$ be given by a set of generators, and let $A, B \in \Lambda(n,q)$ be two alternating matrices. The alternating matrix transporter problem asks to compute the coset $H_{A \to B} = \{g \in H : g^{\mathrm{t}} A g = B\}$.

Theorem 7. *There exists a deterministic algorithm that solves the alternating matrix transporter problem in time* $q^{\frac{1}{4}n^2+O(n)}$.

Theorem 7 is proved using the dynamic programming idea. Details can be found in [33]. Given Theorem 7, we are ready to prove Theorem 3.

Proof: Let $(e_1, \ldots, e_m)$ be the standard basis of $\mathbb{F}_q^m$, and let $E_k = \langle e_1, \ldots, e_k \rangle$. $v = (a_1, \ldots, a_m)^{\mathrm{t}} \in \mathbb{F}_q^m$, define $\mathbf{H}^v := \sum_{i \in [m]} a_i H_i \in \Lambda(n,q)$. For a dimension-$k$ subspace $V \le \mathbb{F}_q^m$ with an ordered basis $(v_1, \ldots, v_k)$, $\mathbf{H}^V := (\mathbf{H}^{v_1}, \ldots, \mathbf{H}^{v_k}) \in \Lambda(n,q)^k$.

The dynamic programming table is indexed by subspaces of $\mathbb{F}_q^m$, so the number of cells is no more than $q^{\frac{1}{4}m^2+O(m)}$. The cell corresponding to a dimension-k subspace V stores the coset

$$\{(g,h) \in \mathrm{GL}(n,q) \times \mathrm{GL}(k,q) : g^{\mathrm{t}} (\mathbf{G}^{E_k}) g = (\mathbf{H}^V)^h\} \quad (1)$$

which is denoted by $\mathrm{Iso}(\mathbf{G}^{E_k}, \mathbf{H}^V)$.

We will fill in the dynamic programming table in the increasing order of the dimension d. Recall that each subspace also comes with an ordered basis by Observation 5. The base case $d = 0$ is trivial. Now assume we have computed $\mathrm{Iso}(\mathbf{G}^{E_\ell}, \mathbf{H}^V)$ for all $1 \le \ell \le d-1$ and $V \le \mathbb{F}_q^n$ of dimension ℓ. To compute $\mathrm{Iso}(\mathbf{G}^{E_d}, \mathbf{H}^V)$ for $V \le \mathbb{F}_q^n$ of dimension d, note that any h in $(g,h) \in \mathrm{Iso}(\mathbf{G}^{E_d}, \mathbf{H}^V)$ satisfies the following. Firstly, h sends E_{d-1} to some dimension-$(d-1)$ subspace $V' \le V$, and $(g,h) \in \mathrm{Iso}(\mathbf{G}^{E_{d-1}}, \mathbf{H}^{V'})$. Secondly, h sends e_k to some $v \in V \setminus V'$, and g sends $\mathbf{G}^{e_d}$ to $\mathbf{H}^v$. This shows that

$$\mathrm{Iso}(\mathbf{G}^{E_d}, \mathbf{H}^V) = \bigcup_{V' \le V, \dim(V')=d-1} \bigcup_{v \in V \setminus V'} \mathrm{Iso}'(V', v),$$

where

$$\mathrm{Iso}'(V', v) = \big[[\mathrm{Iso}(\mathbf{G}^{E_{d-1}}, \mathbf{H}^{V'})](e_d \to v)\big](\mathbf{G}^{e_d} \to \mathbf{H}^v).$$

To compute $\mathrm{Iso}'(V', v)$, $\mathrm{Iso}(\mathbf{G}^{E_{d-1}}, \mathbf{H}^{V'})$ can be read from the table. $[\mathrm{Iso}(\mathbf{G}^{E_{d-1}}, \mathbf{H}^{V'})](e_d \to v)$ is an instance of the pointwise transporter problem of $\mathrm{GL}(n,q) \times \mathrm{GL}(k,q)$ acting on $\mathbb{F}_q^m$, which can be solved in time $q^{O(m)}$. Finally $\big[[\mathrm{Iso}(\mathbf{G}^{E_{d-1}}, \mathbf{H}^{V'})](e_d \to v)\big](\mathbf{G}^{e_d} \to \mathbf{H}^v)$ is an instance of the alternating matrix transporter problem, which can be solved, by Theorem 7, in time $q^{\frac{1}{4}n^2+O(n)}$. Going over the two unions adds a multiplicative factor of q^{2d}, and then we apply Sims' method to reduce the generating set size to $q^{O(n)}$. Therefore for each cell the time complexity is $q^{2d} \cdot q^{\frac{1}{4}n^2+O(n+m)} = q^{\frac{1}{4}n^2+O(m+n)}$. Therefore the whole dynamic programming table can be filled in in time $q^{\frac{1}{4}m^2+O(m)} \cdot q^{\frac{1}{4}n^2+O(n+m)} = q^{\frac{1}{4}(n^2+m^2)+O(n+m)}$. ∎

IV. Discussions and future directions

A. Discussion on the prospect of worst-case time complexity of AltMatSpIso

While our main result is an average-case algorithm, we believe that the ideas therein suggest that an algorithm for AltMatSpIso in time $q^{O(n^{2-\epsilon})}$ may be within reach.

For this, we briefly recall some fragments of the history of GraphIso, with a focus on the worst-case time complexity aspect. Two (families of) algorithmic ideas have been most responsible for the worst-case time complexity improvements for GraphIso. The first idea, which we call the combinatorial idea, is to use certain combinatorial techniques including individualisation, vertex or edge refinement, and more generally the Weisfeiler-Leman refinement [48]. The second idea, which we call the group theoretic idea, is to reduce GraphIso to certain problems in permutation group algorithms, and then settle those problems using group theoretic techniques and structures. A major breakthrough utilising the group theoretic idea is the polynomial-time algorithm for graphs with bounded degree by Luks [35].

Some combinatorial techniques have been implemented and used in practice [39], though the worst-case analysis usually does not favour such algorithms (see e.g. [14]). On the other hand, while group theoretic algorithms for GraphIso more than often come with a rigorous analysis, such algorithms usually only work with a restricted family of graphs (see e.g. [35]). The major improvements on the worst-case time complexity of GraphIso almost always rely on both ideas. The recent breakthrough, a quasipolynomial-time algorithm for GraphIso by Babai [2, 3], is a clear evidence. Even the previous record, a $2^{\tilde{O}(\sqrt{n})}$-time algorithm by Babai and Luks [7], relies on both Luks' group theoretic framework

[35] and Zemlyachenko's combinatorial partitioning lemma [51].

Let us return to ALTMATSPISO. It is clear that ALTMATSPISO can be studied in the context of matrix groups over finite fields. Computing with finite matrix groups though, turns out to be much more difficult than working with permutation groups. The basic constructive membership testing task subsumes the discrete log problem, and even with a number-theoretic oracle, a randomised polynomial-time algorithm for constructive membership testing was only recently obtained by Babai, Beals and Seress [4] for odd q. However, if a $q^{O(n+m)}$-time algorithm for ALTMATSPISO is the main concern, then we can view $\mathrm{GL}(n,q)$ acting on the domain $\mathbb{F}_q^n$ of size q^n, so basic tasks like constructive membership testing are not a bottleneck. In addition, a group theoretic framework for matrix groups in vein of the corresponding permutation group results in [35] has also been developed by Luks [37]. Therefore, if we aim at a $q^{O(n+m)}$-time algorithm for ALTMATSPISO, the group theoretic aspect is relatively developed.

Despite all the results on the group theoretic aspect, as described in Section I-B, a $q^{O(n+m)}$-time algorithm for ALTMATSPISO has been widely regarded to be very difficult, as such an algorithm would imply an algorithm that tests isomorphism of p-groups of class 2 and exponent p in time polynomial in the group order. Reflecting back on how the time complexity of GRAPHISO has been improved, we realised that the other major idea, namely the combinatorial refinement idea, seemed missing in the context of ALTMATSPISO. By adapting the individualisation technique, developing an alternative route to the refinement step as used in [5], and demonstrating its usefulness in the linear algebraic Erdős-Rényi model, we believe that this opens the door to systematically examine and adapt such combinatorial refinement techniques for GRAPHISO to improve the worst-case time complexity of ALTMATSPISO. We mention one possibility here. In [43], a notion of degree for alternating matrix spaces will be introduced, and it will be interesting to combine that degree notion with Luks' group theoretic framework for matrix groups [37] to see whether one can obtain a $q^{O(n+m)}$-time algorithm to test isometry of alternating matrix spaces with bounded degrees. If this is feasible, then one can try to develop a version of Zemlyachenko's combinatorial partition lemma for ALTMATSPISO in the hope to obtain a moderately exponential-time algorithm (e.g. in time $q^{O(n^{2-\epsilon})}$) for ALTMATSPISO.

B. Discussion: on the linear algebraic Erdős-Rényi model

As far as we are aware, the linear algebraic Erdős-Rényi model (Model 1) has not been discussed in the literature. We believe that this model may lead to some interesting mathematics. In this section we put some general remarks on this model. We will consider $\text{LINER}(n,m,q)$, or the corresponding bipartite version of LINER, $\text{BIPLINER}(n \times n, m, q)$, which is the uniform distribution over all m-dimensional subspaces of $M(n,q)$.

To start with, it seems to us reasonable to consider an event E as happening with high probability only when $\Pr[E] \geq 1 - 1/q^{\Omega(n)}$. To illustrate the reason, consider $\text{BIPLINER}(n \times n, m, q)$ with the following property $E(n,m,q)$. For a dimension-m $\mathcal{B} \leq M(n,q)$, $\mathcal{B}$ satisfies $E(n,m,q)$ if and only if for every $U \leq \mathbb{F}_q^n$, $\dim(\mathcal{B}(U)) \geq \dim(U)$. This corresponds to the concept of semi-stable as in the geometric invariant theory; compare with the stable concept as described in Section II-C. One can think of $\mathcal{B}$ being semi-stable as having a perfect matching [20, 25, 26]. When $m = 1$, $\mathcal{B} = \langle B \rangle$ is semi-stable if and only if B is invertible, so $1 - \frac{1}{q} \geq \Pr[E(n,1,q)] \geq 1 - \frac{1}{q-1}$. On the other hand when $m = 4$, since stable implies semi-stable, from Section II-C we have $\Pr[E(n,4,q)] \geq 1 - \frac{1}{q^{\Omega(n)}}$. So though $E(n,1,q)$ happens with some nontrivial probability, it seems not fair to consider $E(n,1,q)$ happens with high probability, while $E(n,4,q)$ should be thought of as happening with high probability.

The above example suggests that the phenomenon in the linear algebraic Erdős-Rényi model can be different from its classical correspondence. Recall that in the classical Erdős-Rényi model, an important discovery is that most properties E have a threshold m_E. That is, when the edge number m is slightly less than m_E, then E almost surely does not happen. On the other hand, if m surpasses m_E slightly then E almost surely happens. m_E is usually a nonconstant function of the vertex number, as few interesting things can happen when we have only a constant number of edges. However, the above example about the semi-stable property suggests that, if there is a threshold for this property, then this threshold has to be between 1 and 4, as we have seen the transition from $1 - 1/q^{O(1)}$ to $1 - 1/q^{\Omega(n)}$ when m goes from 1 to 4. This is not surprising though, as one "edge" in the linear setting is one matrix, which seems much more powerful than an edge in a graph. It should be possible to pin down the exact threshold for the semi-stable property, and we conjecture that the transition (from $1 - 1/q^{O(1)}$ to $1 - 1/q^{\Omega(n)}$) happens from 2 to 3 as this is where the transition from tame to wild as in the representation theory [10, Chapter 4.4] happens for the representations of the Kronecker quivers. This hints on one research direction on LINER, that is, to determine whether the threshold phenomenon happens with monotone properties.

The research on LINER has to depend on whether there are enough interesting properties of matrix spaces. We mention two properties that originate from existing literature; more properties can be found in the forthcoming paper [43]. Let $\mathcal{G}$ be an m-alternating space in $\Lambda(n,q)$. For $U \leq \mathbb{F}_q^n$ of dimension d with an ordered basis $(v_1, \ldots, v_d)$, the restriction of $\mathcal{G}$ to U is defined as $\{[v_1, \ldots, v_d]^t G [v_1, \ldots, v_d] : G \in \mathcal{G}\}$ which is an alternating space in $\Lambda(d,q)$. The first property is the following. Let $s(\mathcal{G})$ be the smallest number

for the existence of a dimension-s subspace U such that the restriction of $\mathcal{G}$ to U is of dimension m. This notion is one key to the upper bound on the number of p-groups [11, 46]. It is interesting to study the asymptotic behavior of $s(\mathcal{G})$. The second property is the following. Call $U \leq \mathbb{F}_q^n$ an independent subspace, if the restriction of $\mathcal{G}$ to U is the zero space. We can define the independent number of $\mathcal{G}$ accordingly. This mimics the independent sets for graphs, and seems to relate to the independent number concept for non-commutative graphs which are used to model quantum channels [15]. Again, it is interesting to study the asymptotic behavior of the independent number.

Finally, as suggested in [15] (where they consider Hermitian matrix spaces over $\mathbb{C}$), the model may be studied over infinite fields, where we replace "with high probability" with "generic" as in the algebraic geometry sense.

ACKNOWLEDGMENT

We thank Gábor Ivanyos and James B. Wilson for helpful discussions and useful information. Y. Q. was supported by Australian Research Council DECRA DE150100720 during this work.

REFERENCES

[1] László Babai. Monte-Carlo algorithms in graph isomorphism testing. Technical Report 79-10, Dép. Math. et Stat., Université de Montréal, 1979.

[2] László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 684–697, 2016. arXiv:1512.03547, version 2.

[3] László Babai. Fixing the UPCC case of Split-or-Johnson. http://people.cs.uchicago.edu/ laci/upcc-fix.pdf, 2017.

[4] László Babai, Robert Beals, and Ákos Seress. Polynomial-time theory of matrix groups. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 55–64, 2009.

[5] László Babai, Paul Erdős, and Stanley M. Selkow. Random graph isomorphism. *SIAM J. Comput.*, 9(3):628–635, 1980.

[6] László Babai and Ludek Kučera. 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.

[7] László Babai and Eugene M. Luks. Canonical labeling of graphs. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 171–183, 1983.

[8] László Babai and Youming Qiao. Polynomial-time isomorphism test for groups with Abelian Sylow towers. In *29th STACS*, pages 453 – 464. Springer LNCS 6651, 2012.

[9] Reinhold Baer. Groups with abelian central quotient group. *Transactions of the American Mathematical Society*, 44(3):357–386, 1938.

[10] David J. Benson. Representations and cohomology. i, volume 30 of cambridge studies in advanced mathematics, 1998.

[11] Simon R. Blackburn, Peter M. Neumann, and Geetha Venkataraman. *Enumeration of finite groups*. Cambridge Univ. Press, 2007.

[12] Peter A. Brooksbank, Joshua Maglione, and James B. Wilson. A fast isomorphism test for groups of genus 2. arXiv:1508.03033, 2015.

[13] Peter A. Brooksbank and James B. Wilson. Computing isometry groups of Hermitian maps. *Trans. Amer. Math. Soc.*, 364:1975–1996, 2012.

[14] Jin-yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identifications. *Combinatorica*, 12(4):389–410, 1992.

[15] Runyao Duan, Simone Severini, and Andreas J. Winter. Zero-error communication via quantum channels, noncommutative graphs, and a quantum lovász number. *IEEE Trans. Information Theory*, 59(2):1164–1174, 2013.

[16] Bettina Eick, C. R. Leedham-Green, and E. A. O'Brien. Constructing automorphism groups of p-groups. *Communications in Algebra*, 30(5):2271–2295, 2002.

[17] Paul Erdős and Alfréd Rényi. Asymmetric graphs. *Acta Mathematica Hungarica*, 14(3-4):295–315, 1963.

[18] V. Felsch and J. Neubüser. On a programme for the determination of the automorphism group of a finite group. In Pergamon J. Leech, editor, *Computational Problems in Abstract Algebra (Proceedings of a Conference on Computational Problems in Algebra, Oxford, 1967)*, pages 59–60, Oxford, 1970.

[19] François Le Gall and David J. Rosenbaum. On the group and color isomorphism problems. *CoRR*, abs/1609.08253, 2016.

[20] Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 109–117, 2016.

[21] Joshua A. Grochow and Youming Qiao. Algorithms for group isomorphism via group extensions and cohomology. *SIAM Journal on Computing*, 46(4):1153–1216, 2017.

[22] Joshua A. Grochow and Youming Qiao. Isomorphism

problems in linear algebra. In preparation, 2017.

[23] Graham Higman. Enumerating p-groups. I: Inequalities. *Proceedings of the London Mathematical Society*, 3(1):24–30, 1960.

[24] 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. arXiv: 1708.03495, 2017.

[25] Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Non-commutative Edmonds' problem and matrix semi-invariants. *Computational Complexity*, pages 1–47, 2016.

[26] Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Constructive non-commutative rank is in deterministic polynomial time. In *the 8th Innovations in Theoretical Computer Science (ITCS)*, 2017.

[27] William M. Kantor and Eugene M. Luks. Computing in quotient groups. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 524–534, 1990.

[28] Richard M. Karp. Probabilistic analysis of a canonical numbering algorithm for graphs. In *Proceedings of the AMS Symposium in Pure Mathematics*, volume 34, pages 365–378, 1979.

[29] Dmitry Kerner and Victor Vinnikov. Determinantal representations of singular hypersurfaces in pn. *Advances in Mathematics*, 231(3):1619–1654, 2012.

[30] Johannes Köbler, Uwe Schöning, and Jacobo Torán. *The graph isomorphism problem: its structural complexity*. Birkhauser Verlag, Basel, Switzerland, Switzerland, 1993.

[31] Mark L. Lewis and James B. Wilson. Isomorphism in expanding families of indistinguishable groups. *Groups - Complexity - Cryptology*, 4(1):73110, 2012.

[32] Yinan Li and Youming Qiao. Almost sharp bound on the number of p-groups of Frattini class 2 of a given order. In preparation, 2017.

[33] Yinan Li and Youming Qiao. Linear algebraic analogues of the graph isomorphism problem and the Erdős-Rényi model. arXiv:1708.04501, 2017.

[34] Richard J. Lipton. The beacon set approach to graph isomorphism. Yale University. Department of Computer Science, 1978.

[35] Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comput. Syst. Sci.*, 25(1):42–65, 1982.

[36] Eugene M. Luks. Lectures on polynomial-time computation in groups. *Lecture notes*, 1990.

[37] Eugene M. Luks. Computing in solvable matrix groups. In *33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, Pennsylvania, USA, 24-27 October 1992*, pages 111–120, 1992.

[38] 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 '99, pages 652–658, New York, NY, USA, 1999. ACM.

[39] Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, II. *Journal of Symbolic Computation*, 60(0):94 – 112, 2014.

[40] Gary L. Miller. On the $n^{\log n}$ isomorphism technique (a preliminary report). In *STOC*, pages 51–58, New York, NY, USA, 1978. ACM.

[41] David Mumford, John Fogarty, and Frances Kirwan. *Geometric invariant theory*. Springer-Verlag, 1994.

[42] Eamonn A. O'Brien. Isomorphism testing for p-groups. *Journal of symbolic computation*, 16(3):305–320, 1993.

[43] Youming Qiao. Matrix spaces as a linear algebraic analogue of graphs. In preparation, 2017.

[44] David J. Rosenbaum. Bidirectional collision detection and faster deterministic isomorphism testing. *arXiv preprint arXiv:1304.3935*, 2013.

[45] Ákos Seress. *Permutation group algorithms*, volume 152. Cambridge University Press, 2003.

[46] Charles C. Sims. Enumerating p-groups. *Proceedings of the London Mathematical Society*, 3(1):151–166, 1965.

[47] Robert B. Warfield. *Nilpotent Groups*. Number 513 in Lecture Notes in Mathematics; 513. Springer-Verlag, 1976.

[48] Boris Weisfeiler and Andrei A. Leman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, 2(9):12–16, 1968.

[49] James B. Wilson. Decomposing p-groups via Jordan algebras. *Journal of Algebra*, 322(8):2642–2679, 2009.

[50] 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.

[51] Viktor N. Zemlyachenko, Nickolay M. Korneenko, and Regina I. Tyshkevich. Graph isomorphism problem. *Journal of Soviet Mathematics*, 29(4):1426–1481, 1985.