

Incorporating Weisfeiler–Leman into algorithms for group isomorphism

Peter A. Brooksbank* Joshua A. Grochow† Yinan Li‡ Youming Qiao§

James B. Wilson¶

May 8, 2019

Abstract

In this paper we combine many of the standard and more recent algebraic techniques for testing isomorphism of finite groups (Gpl) with combinatorial techniques that have typically been applied to Graph Isomorphism. We show how to combine several state-of-the-art Gpl algorithms for specific group classes into an algorithm for general Gpl, namely: composition series isomorphism (Rosenbaum–Wagner, *Theoret. Comp. Sci.*, 2015; Luks, 2015), recursively-refineable filters (Wilson, *J. Group Theory*, 2013), and low-genus Gpl (Brooksbank–Maglione–Wilson, *J. Algebra*, 2017). Recursively-refineable filters—a generalization of subgroup series—form the skeleton of this framework, and we refine our filter by building a hypergraph encoding low-genus quotients, to which we then apply a hypergraph variant of the k -dimensional Weisfeiler–Leman technique. Our technique is flexible enough to readily incorporate additional hypergraph invariants or additional characteristic subgroups as they emerge.

After introducing this general technique, we prove three main results about its complexity:

- Let the *width* of a filter be the dimension of the largest quotient of two adjacent subgroups of the filter; the *color-ratio* of our hypergraph captures how much smaller a color class is compared to the layer of the filter it is coloring. When we use genus- g quotients and hypergraph k -WL, we can solve isomorphism for solvable groups of order n in time

$$\left(\frac{n}{\text{color-ratio}}\right)^{\text{width}} \text{poly}(n) + n^{O(gk)}$$

In the “base case”, where the solvable radical is itself low-genus and the semisimple part acts trivially, we can get a better guaranteed running time of $n^{O(\log \log n)}$, by combining cohomological techniques (Grochow–Qiao, CCC '14, *SIAM J. Comput.*, 2017), code equivalence (Babai–Codenotti–Grochow–Qiao, SODA '11), and low-genus isomorphism ([BMW], *ibid.*).

- We introduce a new random model of finite groups. Unlike previous models, we prove that our model has good coverage, in that it produces a wide variety of groups, and in particular a number of distinct isomorphism types that is logarithmically equivalent to the number of all isomorphism types. In this random model, we show that our filter-and-1-WL refinement method results in constant *average* width (the above result uses max width).
- For p -groups of class 2 and exponent p —widely believed to be the hardest cases of Gpl, and where we also expect the above techniques to get stuck—we improve on the average-case algorithm of Li–Qiao (FOCS '17). Our new algorithm is simpler and applies to a larger fraction of random p -groups of class 2 and exponent p . The previous algorithm was based on a linear-algebraic analogue of the individualize-and-refine technique; our new algorithm combines that technique with concepts from isomorphism of low-genus groups. We also implement this algorithm in MAGMA and show that in experiments it improves over the default (brute force) algorithm for this problem.

*Department of Mathematics, Bucknell University, Lewisburg, PA 17837, United States. pbrooks@bucknell.edu

†Departments of Computer Science and Mathematics, University of Colorado—Boulder, Boulder, CO 80309-0430, United States. jgrochow@colorado.edu

‡CWI and QuSoft, Science Park 123, 1098XG Amsterdam, Netherlands. Yinan.Li@cwi.nl.

§Center for Quantum Software and Information, University of Technology Sydney, Ultimo NSW 2007, Australia. Youming.Qiao@uts.edu.au

¶Department of Mathematics, Colorado State University, Fort Collins, CO 80523, United States. James.Wilson@ColoState.Edu

1 Introduction

The problem of deciding whether two finite groups are isomorphic (**Gpl**) has a century-old history that straddles several fields, including topology, computational algebra, and computer science. It also has several unusual variations in complexity. For example, the dense input model—where groups are specified by their multiplication “*Cayley*” tables, has quasi-polynomial time complexity and it reduces *to* the better known *Graph Isomorphism* problem (**Gphl**); cf. [L3, Section 10]. Meanwhile, **Gpl** for a sparse input model for groups, such as by permutations, matrices, or black-box groups, reduces *from* **Gphl**; cf. [HL, LV]. At present sparse **Gpl** is in $\Sigma_2\text{P}$ and it is not known to lie in either NP nor coNP; see [BS, Propostion 4.8, Corollary 4.9]. In fact in the model of groups input by generators and relations, Adian and Rabin famously showed **Gpl** is undecidable [A, R1].

Following L. Babai’s breakthrough proof that **Gphl** is in quasi-polynomial time [B1], dense (Cayley table) **Gpl** is now an essential bottleneck to improving graph isomorphism. So while the methods we explore here can be applied in both the dense and the sparse models of **Gpl**, we concentrate our complexity claims on the dense case. In particular, when we say polynomial-time, we mean polynomial time in the group order unless specified otherwise. Our contribution here is to expose from within the structure of groups, graph theoretic properties which relate to the difficulty of solving **Gpl**. We expect this to facilitate the systematic use of combinatorial isomorphism techniques within **Gpl** that interplay with existing algebraic strategies.

We introduce a colored (hyper-)graph based on an algebraic data structure known as a *recursively-refineable filter* which identifies abelian groups and vector spaces layered together to form the structure of a finite group. Filters have been useful in several isomorphism tests [W3, M3, M1, BOW]. As the name suggests, filters can be refined, and with each refinement the cost $T(n)$ of isomorphism testing decreases after refinement to a function in $O(T(n)^{1/c})$, for some $c \geq 2$. The more rounds of refinement we can carry out the lower the cost of isomorphism. Existing uses of filter refinement find characteristic structure algebraically; our principal innovation is to add a combinatorial perspective to refinement. We color (co)dimension- g subspaces of the layers of the filter using local isomorphism invariants. This parameter g will be referred to as the *genus parameter*. The layers are in turn connected to each other according to their position within the group, and this presents further opportunities for coloring. With so much nuanced local information, the graph we associate to a group is well suited to individualization and refinement techniques like the dimension- k Weisfeiler–Leman procedure [WL, B2, IL, CFI]. The critical work is to refine these graphs compatibly with the refinement of the filter (Theorem A). Thus, one maintains the relationship between the group and graph isomorphism properties as we recursively refine. While our methods do not apply to structures as general as semigroups and quaisgroups, they can be adapted to other problems, such as ring isomorphism [KS].

To explore the implications of this technique we introduce a model for random finite groups. In doing so we consider pitfalls identified in previously suggested models for random finite groups. We are especially concerned with *coverage*—the idea that we are able to easily sample from groups within natural classes such as non-solvable, solvable, nilpotent, and abelian—and that within each subclass the number of isomorphism types is dense on a log scale (Theorem C). Log-scale is for now the best granularity we know for the enumeration of groups, cf. [BNV]. We then prove (Theorem D) that in our random model, genus-1 1-WL-refinement on average refines to a series of length $\Theta(\log n)$, which thereby achieves the expected refinement length posed in [W3, p. 876]. Following the refinement, the *average* width of the filter is thus constant, though the cost of refinement increases. (If the maximum width were constant it would result in a polynomial-time average-case isomorphism test in our model.)

Finally within our random model there are several “base cases” where the recursive refinements

become less likely, or where our analysis is inadequate. We demonstrate that in two of these cases, isomorphism can be solved either in polynomial time in the average-case sense (Theorem E) or in nearly polynomial time ($n^{O(\log \log n)}$) in the worst-case sense (Theorem F). The former also solves a related problem of average complexity of tensor equivalence.

Our strategy harnesses critical features of a great variety of existing approaches to isomorphism (code equivalence, filter refinements, adjoint-tensor methods, bidirectional collision) and uses Weisfeiler–Leman refinement as the top-level strategy to combine the various implications. That diversity was not so much a plan but the result of hitting barriers and looking to the literature for solutions. The result, however, is a framework that is rather flexible and is well suited to accommodate future ideas, both algebraic and combinatorial, as featured here already. That strength of course comes at a cost that the mechanics and analysis are rather involved. We expect that in time better analysis and simplified models will improve our understanding.

1.1 The context of this work

Much recent progress in Gpl has been had by considering special classes of groups; the recent papers [BMW1, GQ, LGR] survey and supplement these results. That has created powerful but highly varied strategies with no obvious means of synthesis. Within our refinement model of computing Gpl we have the opportunity to begin merging some of the many options that have been developed to date. To help explain our approach we consider examples of groups of invertible matrices over finite fields of prime order, as graphically communicated in Figure 1.1. In fact, these examples will later evolve into the aforementioned random model for finite groups.

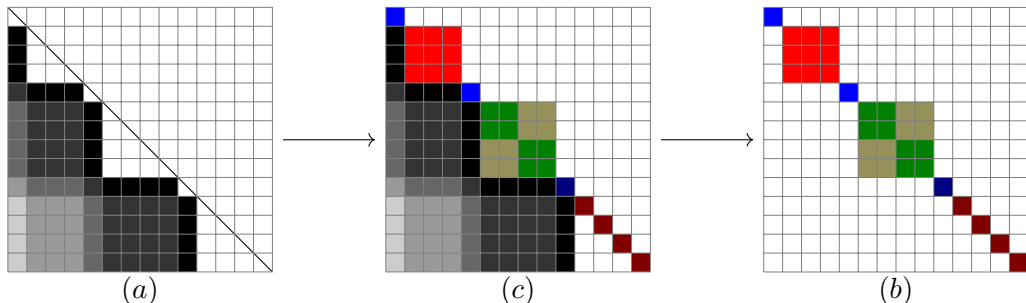


Figure 1.1: Diagrams of matrix groups can capture many of the well-studied examples of finite groups: (a) depicts a large variety of nilpotent groups; (b) depicts products of quasi- and almost-simple groups together with possible permutations of isomorphic blocks; and (c) depicts wide range of general finite groups decomposed into smaller classes of groups.

First thread: connection with linear and multilinear algebras. Algorithms and data structures for linear and multilinear structures are on the whole far more evolved than counterparts for groups. This explains why progress for groups can be made by mapping problems into the realm of linear and multilinear algebra. Such a correspondence has been known for close to a century, originating in work of Brahana [B3] and Baer [B3]. Consider groups U of the following form.

$$U \leq H(d_1, \dots, d_\ell; \mathbb{F}) := \left\{ \left[\begin{array}{cccc} I_{d_1} & a_{12} & a_{22} & \cdots \\ & I_{d_2} & a_{23} & \\ & & \ddots & \ddots \\ & & & I_{d_\ell} \end{array} \right] \mid a_{ij} \in M_{d_i \times d_j}(\mathbb{F}) \right\}.$$

Figure 1.1(a) illustrates a possible U . In the creation of our random model we shall sample groups U by selecting random matrices in $H(d_1, \dots, d_\ell; \mathbb{F})$. A surprising necessity is that we sample only sparse matrices. Although this might seem counter to the goals of seeding a group with lots of entropy, we will demonstrate that groups with too much random seeding become virtually identical (Theorem 5.2).

As a general remark it will be helpful throughout this work to regard all groups $U = \langle U, \cdot, {}^{-1}, 1 \rangle$ as having been enriched by the addition of a second binary operation $[a, b] = a^{-1}b^{-1}ab$ known as *commutation*. In this way groups behave much more like rings than they do like semigroups or quasigroups. In particular, $[,]$ very nearly distributes over the usual binary operation \cdot in U , in that $[ab, c] = b^{-1}[a, c]b[b, c]$. That explains the link to multilinear algebra. In the case of U :

$$\left[\left[\begin{array}{ccc} I_{d_1} & a_{12} & \cdots \\ & \ddots & \ddots \\ & & I_{d_\ell} \end{array} \right], \left[\begin{array}{ccc} I_{d_1} & b_{12} & \cdots \\ & \ddots & \ddots \\ & & I_{d_\ell} \end{array} \right] \right] = \left[\begin{array}{ccc} I_{d_1} & a_{12} + b_{12} & a_{22} + b_{22} + a_{12}b_{23} - b_{12}a_{23} & \cdots \\ & \ddots & & \ddots \\ & & & I_{d_\ell} \end{array} \right].$$

Stripping away the addition leaves us to compare bilinear (and later multilinear) products such as $(a_{ij}, b_{jk}) \mapsto a_{ij}b_{jk}$, under base changes. We treat these as functions $\mathbb{F}^a \times \mathbb{F}^b \mapsto \mathbb{F}^c$, where \mapsto indicates the function is multilinear. Equivalently, we must study the orbits of groups $\mathrm{GL}(a, \mathbb{F}) \times \mathrm{GL}(b, \mathbb{F}) \times \mathrm{GL}(c, \mathbb{F})$ acting on elements of the tensor space $\mathbb{F}^a \otimes \mathbb{F}^b \otimes \mathbb{F}^c$. Such reductions of group isomorphism to multilinear equivalence, and more general *tensor equivalence* problems have been the key to the recent progress on some of the largest and most difficult instances of **Gpl** [**BMW1, BW, LQ, IQ, LW2, BOW, W2**]. The strategies buried within those methods are nevertheless quite distinct. For example, several focus on $*$ -algebras and properties of rings and modules acting on tensors. Others focus on tensors as high-dimensional arrays, and perform individualization and refinement techniques on slices of this data structure. Our model of refinement allows for both strategies.

Second thread: relationship to code equivalence. Now consider the types of groups we could place on the block diagonal of the matrix group examples in Figure 1.1. These could include groups like $\mathrm{GL}(d_i, \mathbb{F})$. We could also use subgroups such as $\mathrm{GL}(1, \mathbb{F}_{p^{d_i}}) = \mathbb{F}_{p^{d_i}}^\times$, as well as natural families of geometrically interesting groups such as orthogonal, unitary, and symplectic groups. We may even embed the same group several times into multiple blocks on the diagonal, e.g. $\langle \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} : A \in \mathrm{GL}(e, \mathbb{F}) \rangle$. Those blocks could further be permuted producing groups of block monomial matrices such as $\langle \begin{bmatrix} 0 & A \\ A & 0 \end{bmatrix} : A \in \mathrm{GL}(e, \mathbb{F}) \rangle$. We can capture the spirit of such a group graphically in Figure 1.1(b). Indeed, our random group model builds random semi-simple and quasi-semisimple groups in just this way.

Isomorphism in the context of groups of this kind has been approached mostly through the use of code equivalence. For example, for semisimple groups—those with no non-trivial abelian normal subgroups—there is an algorithm that runs in time polynomial in the group order [**BCGQ, BCQ**], as well as an algorithm that is efficient in practice [**CH**]. The key algorithmic idea is dynamic programming, and its use follows the one by Luks in the context of hypergraph isomorphism and code equivalence [**L2**]. Later [**GQ**] considers the further implications when the groups centrally extend abelian groups similar to the general family we have described in this thread.

Third thread: composition series and filters. In recent years there has been some progress on improving general isomorphism using subgroup chains. Rosenbaum and Wagner [**RW**] demonstrated that one can fix a composition series $\mathcal{C}(G)$ for group G and then, given a composition series $\mathcal{C}(H)$ for another group H , efficiently decide if there is an isomorphism $G \rightarrow H$ sending $\mathcal{C}(G)$ to $\mathcal{C}(H)$.

Luks gave an improvement of that test [L1]. In this way, the putative cost of $n^{\log n + O(1)}$ steps to decide isomorphism by brute-force is reduced to the number of possible composition series, which is at most $n^{\binom{1}{2} + o(1)} \log n$.

Filters can use characteristic subgroups to recursively find more characteristic subgroups, ultimately producing a large enough collection of fixed subgroups that an isomorphism test along the lines of Rosenbaum–Wagner becomes efficient. For several families of groups such refinements have been discovered [W3, M2]. Our approach here extends the filtration process by taking the methods known and combining them into a colored hypergraph where individualization–refinement techniques can be applied. The goal is to make it even more likely to reach a situation in which the Rosenbaum–Wagner and Luks algorithms can be applied efficiently.

1.2 An outline of the Weisfeiler–Leman procedure for groups

Our approach to `Gpl` uses recursively-refineable filters to build and refine a colored hypergraph within, and between, abelian layers of a given group. A filter ϕ on a group G assigns to a c -tuple $s = (s_1, \dots, s_c)$ of natural numbers (including 0) a normal subgroup ϕ_s of G subject to natural compatibility requirements. Let $\text{Norm}(G)$ denote the set of normal subgroups of G , and for $A, B \subseteq G$ let $[A, B] = \langle [a, b] \mid a \in A, b \in b \rangle$.

Definition 1.1 (Filter [W3]). A *filter* on a group G is a map $\phi: \mathbb{N}^d \rightarrow \text{Norm}(G)$, where

$$(\forall s, t \in \mathbb{N}^d) \quad s \leq_{\text{lex}} t \implies \phi_s \geq \phi_t \text{ and } [\phi_s, \phi_t] \leq \phi_{s+t}. \quad (1.2)$$

Note that the first condition implies that the subgroups ϕ_s form a descending chain of subgroups, though in general it is not a proper chain. Computationally we only store the lexicographically least label s for each distinct subgroup ϕ_s in the image of ϕ . Thus, a filter’s image is bounded by the length of the longest subgroup chain. For a group of order n this is at most $\log n$.

We begin with a filter $\phi: \mathbb{N}^c \rightarrow \text{Norm}(G)$ known from the structure of general finite groups, and then refine by increasing the value of c . That refinements exist is proved in [W3] and that they can be computed efficiently is shown in [M3]. Our initial value for c will be the number of distinct primes p_1, \dots, p_c dividing $n = |G|$. For each prime p_i , we let $O_{p_i}(G)$ denote the intersection of all Sylow p_i -subgroups of G , the maximum normal subgroup having order a power of p_i . Let $e_i = (\dots, 0, \underset{i}{1}, 0, \dots) \in \mathbb{N}^c$, sorted lexicographically (so that $e_i < e_{i+1}$), and define $\phi: \mathbb{N}^c \rightarrow \text{Norm}(G)$ as follows:

$$\phi_s = \begin{cases} G & s = 0, \\ \prod_{j=i}^c O_{p_j}(G) & s = e_i, \\ [\phi_{s_i e_i}, G] \phi_{s_i e_i}^{p_i} & s = (s_i + 1) e_i, \\ \prod_{i=1}^c \phi_{s_i e_i} & s = \sum_{i=1}^c s_i e_i. \end{cases}$$

Here the product $\prod_i \phi_{s_i e_i}$ means the normal subgroup generated by the terms $\phi_{s_i e_i}$. For example the group S_4 of permutations on 4 letters would have

$$\phi_{(0,0)} = S_4 \geq \phi_{(1,0)} = O_2(S_4)O_3(S_4) = \langle (12)(34), (13)(24) \rangle \geq \phi_{(2,0)} = \phi_{(0,1)} = O_3(S_4) = 1.$$

The *boundary filter* $\partial\phi: \mathbb{N}^d \rightarrow \text{Norm}(G)$ is defined by $\partial\phi_s = \langle \phi_{s+t} : t \in \mathbb{N}^d \setminus \{0\} \rangle$ (if $d = 1$, then $\partial\phi_s = \phi_{s+1}$), and the quotients $L_s := \phi_s / \partial\phi_s$ are the *layers* of ϕ . Note that for each $s \neq 0$, L_s is abelian, and in fact a $\mathbb{Z}[\phi_0 / \partial\phi_0]$ -module. In the selected filter above these are in fact \mathbb{F}_{p_i} -vector spaces for some i . The set $L(\phi) = \bigoplus_{s \neq 0} L_s$, with homogeneous bilinear products

$$[\cdot]_{st} : L_s \times L_t \mapsto L_{s+t} : (x \partial\phi_s, y \partial\phi_t) \mapsto [x, y] \partial\phi_{s+t},$$

is a graded Lie algebra whose graded components are invariant under $\text{Aut}(G)$ [W3, Theorem 3.1].

A bilinear map (bimap) $L_s \times L_t \rightarrow L_{s+t}$ is said to have *genus* g if it is defined over a field \mathbb{F} such that $\dim_{\mathbb{F}} L_{s+t} \leq g$, or (see [BMW1] for details) if it is built from such maps by certain elementary products (such as direct products, but even “central” products are allowed). We will primarily be concerned with the case where $\mathbb{F} = \mathbb{Z}_p$ and we consider bimaps whose codomain has dimension at most g , but our results extend without difficulty to the more general notion of genus.

In our setting the layers of ϕ are elementary abelian, and our approach is to build a hypergraph whose vertices are the union of the points (1-spaces) in the projective geometries of the layers. For $s \in \mathbb{N}^d$, let $\text{PG}_k(L_s)$ denote the set of $(k+1)$ -dimensional subspaces of L_s . Define a family of hypergraphs $\mathcal{H}^{(g)}(\phi)$, where $1 < g \in \mathbb{Z}$ is a parameter, with vertices and hyperedges defined as follows:

The vertex set of $\mathcal{H}^{(g)}(\phi)$ is $\mathcal{V} = \bigcup_{s \in \mathbb{N}^d} \text{PG}_0(L_s)$.

The hyperedge set of $\mathcal{H}^{(g)}(\phi)$ is $\mathcal{E} = \bigcup_{s \in \mathbb{N}^d} \text{PG}_g(L_s) \cup \text{PG}_{\dim L_s - g}(L_s) \cup \bigcup_{s \neq t} \mathcal{K}_{st}$, where \mathcal{K}_{st} is a hypergraph with edges and 3-edges on $\text{PG}_0(L_s) \cup \text{PG}_0(L_t) \cup \text{PG}_0(L_{s+t})$.

Having defined the hypergraph $\mathcal{H}^{(g)}(\phi)$, we shall apply the k -dimensional Weisfeiler-Leman (WL) procedure to it in an appropriate way. Briefly, it is a hypergraph version of the WL procedure [WL] on graphs [B2, IL]. When $k = 1$, such a WL procedure on hypergraphs was recently studied by Böker [B1].

To this end we obtain an algorithm that, given a finite group G and integers $g, k \geq 1$, computes a suitable characteristic filter $\phi: \mathbb{N}^d \rightarrow \text{Norm}(G)$, where $N = O_{\infty}(G) = \prod_p O_p(G)$ is the Fitting subgroup, and an associated hypergraph $\mathcal{H}^{(g,k)}(\phi)$. Further, it colors the hyperedges \mathcal{E} of $\mathcal{H}^{(g,k)}(\phi)$ in a certain desirable way. If $\chi: \mathcal{E} \rightarrow \mathbb{N}$ is a coloring of hyperedges, denote the corresponding colored hypergraph by $\mathcal{H}_{\chi}^{(g,k)}(\phi)$.

Theorem A. *There is a deterministic algorithm that, given a finite group G and integers $g, k \geq 1$, constructs the Fitting subgroup $N = O_{\infty}(G)$, a characteristic filter $\phi: \mathbb{N}^d \rightarrow \text{Norm}(G)$ whose non-zero layers are elementary abelian $\text{Aut}(G)$ -modules, the hypergraph $\mathcal{H} = \mathcal{H}^{(g,k)}(\phi)$, and a coloring $\chi: \mathcal{V}(\mathcal{H}) \cup \mathcal{E}(\mathcal{H}) \rightarrow \mathbb{N}$ satisfying:*

- (i) $\mathcal{H}_{\chi}^{(g,k)}(\phi)$ is hereditary in the following sense: for each $s \in \mathbb{N}^d - \{0\}$, the vertex-and-edge-colored hypergraph obtained by restricting $\mathcal{H}_{\chi}^{(g,k)}(\phi)$ to G/ϕ_s is a refinement of the colored hypergraph for G/ϕ_s based on the filter ϕ truncated at ϕ_s .
- (ii) $\mathcal{H}_{\chi}^{(g,k)}(\phi)$ is also hereditary in k in the following sense: the underlying hypergraphs of $\mathcal{H}^{(g,k)}(\phi)$ and $\mathcal{H}^{(g,k+1)}(\phi)$ are identical, and the coloring of the latter refines the coloring of the former.
- (iii) If $G \cong G'$, there is a colored hypergraph isomorphism $f: \mathcal{H}_{\chi}^{(g,k)}(\phi) \rightarrow \mathcal{H}_{\chi'}^{(g,k)}(\phi')$ such that

$$\begin{aligned} \forall e \in \mathcal{E}(\mathcal{H}), \quad \chi(e) &= \chi'(f(e)), \\ \forall v \in \mathcal{V}(\mathcal{H}), \quad \chi(v) &= \chi'(f(v)). \end{aligned}$$

The time complexity is $|G|^{O(gk)}$.

The algorithm to construct the colored hypergraph $\mathcal{H}_{\chi}^{(g,k)}(\phi)$ is an iterative procedure that we describe in detail in Section 3. Within a fixed iteration, we apply a Weisfeiler–Leman type individualization procedure to obtain a stable coloring (a hypergraph analogue of k -dimensional

WL). We then use that stable coloring to search for characteristic structure in G not already captured by the filter ϕ . If we succeed, we use this structure to refine ϕ and iterate.

Given the result of our WL-algorithm and applying Luks's extension [L1] of the Rosenbaum–Wagner composition series comparison [RW], whenever we refine we improve our isomorphism test, resulting in:

Theorem B. *Let $\phi = \phi_{g,k}$ and $\mathcal{H}^{(g,k)} = \mathcal{H}_\chi^{(g,k)}(\phi)$ denote the filter and colored hypergraph from Theorem A. Let $\text{width}(\phi)$ denote the maximum dimension of any layer of ϕ , and let $\text{color-ratio}(\mathcal{H})$ be the product over all layers s of $|L_s|/|C_s|$, where C_s is the smallest color class in layer L_s . Then given a nilpotent group N of order n , isomorphism can be tested in time*

$$\left(\frac{n}{\text{color-ratio}(\mathcal{H}^{(g,k)})} \right)^{\text{width}(\phi_{g,k})} \text{poly}(n) + n^{O(gk)}.$$

We extend this with an individualize-and-refine technique in Section 4.3, though for that we do not have as cleanly stated an upper bound.

Remark 1.3. The initial filter described above can be extended to solvable groups, and in particular the solvable radical $\text{Rad}(G)$ of any group, by doing something similar to the above within each layer of the Fitting series. This would let us extend all our results from using the Fitting subgroup $O_\infty(G)$ to using the solvable radical $\text{Rad}(G)$ instead, and would extend Theorem B from nilpotent to solvable groups.

1.3 An outline of the random model

Unlike sampling a random graph, where edges can freely be added or omitted, sampling groups of a fixed order requires some delicacy. For example, there are 15 isomorphism types of groups of order 16 but only 1 each of orders 15 and 17. Sampling random groups has hitherto been approached in one of the following two ways.

Quotient Sampling. Fix a free group $F[X]$ of all strings on an alphabet $X \cup X^{-1}$, and consider quotients by normal subgroups $N = \langle S \rangle$ sampled by choosing $S \subset F$ by some aleatory process.

Subgroup Sampling. Fix an automorphism group of a structure, such as the group $\text{Sym}(\Omega)$ of permutations of a set Ω , or the group $\text{GL}(V)$ of invertible linear transformations of a vector space V . Consider subgroups $H = \langle S \rangle$ where S is sampled by some aleatory process.

Evidently, both methods yield groups, but neither offers sufficient variability when restricted to finite groups. For instance Gromov studied quotient sampling as a function of the word lengths of elements in S , finding most quotients are 1, $\mathbb{Z}/2$, or infinite [G2]. Also, subgroup sampling in $G = \text{Sym}(\Omega)$ (respectively $\text{GL}(V)$) has been shown by Dixon, Kantor–Lubotsky [KL], and others to essentially sample A_n , S_n (respectively, subgroups $\text{SL}(V) \leq H \leq \text{GL}(V)$).

To escape these conditions we adopt a method of sampling that appears antithetical to random models: we strongly bias our random selections. We settle on a model related to subgroup sampling in $\text{GL}(d, p)$ since this affords us easy-to-use group operations. (Note, Novikov–Boone demonstrated that the word problem in the free group is undecidable and thus working with quotients $F[X]/N$ is not in general feasible [N, B2].)

First, we sample random upper $(d \times d)$ -unitriangular matrices $u_1, \dots, u_\ell \in U(d, p)$ but we insist that they are ϵ -sparse, for some constant ϵ . Then

$$U = \langle u_1, \dots, u_\ell \rangle.$$

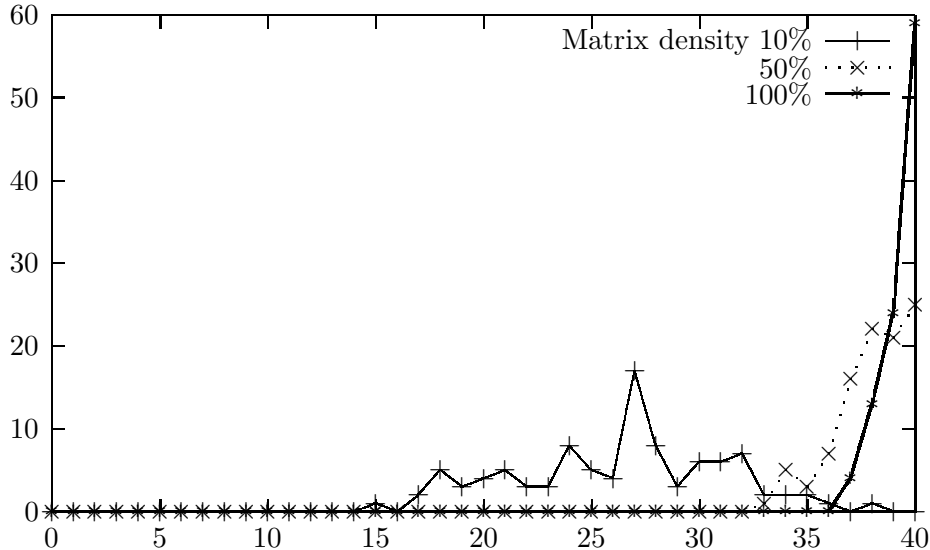


Figure 1.2: Plots of the orders of 100 subgroups sampled as $\langle u_1, \dots, u_5 \rangle \leq U(10, 3)$ with three different densities ϵ : $(+, \epsilon = 1/10)$, $(\times, \epsilon = 1/2)$, and $(*, \epsilon = 1)$. Greater density makes group order, and structure, less varied. The X -axis is labelled by the group order, while the Y -axis is labelled by the percentage of the sampled groups.

samples a subgroup whose order is a power of the prime p characteristic of our fixed field \mathbb{F} . As we shall demonstrate in Theorem 5.2, *without limiting our randomness to sparse matrices* the groups U will almost always contain the following subgroup.

$$\gamma_2(U(d, p)) = \left\{ \begin{bmatrix} 1 & 0 & * & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 0 & * \\ & & & 1 & 0 \\ & & & & 1 \end{bmatrix} \right\}.$$

In essence, this is a p -group analogue of the observations we made about sampling in S_n and $GL(d, p)$. However, sampling with sparsity gives substantial variation, as illustrated simply by comparing orders in Figure 1.2. An interesting recent study by R. Gilman describes a similar situation for permutations analyzed by Kolomogorov complexity [G1].

Secondly, once we have selected a suitably random upper unitriangular group U , an extension to this group is selected by adding to its block-diagonal. That process consists of choosing a partition of the series of common generalized eigen 1-spaces (the fixed point flag) of the group U . In each block we select a random (almost) quasisimple group with a representation of dimension at most the size of the block. We further allow for multiplicity and for permutations of isomorphic modules. This extends U first by a block-diagonal abelian group, then a product of simple groups, followed by a layer of abelian groups, and a final layer of permutations. It is well known that every finite group has such a decomposition, often referred to as the *Babai–Beals* filtration [BB]. We note our own filtration descends to the Fitting subgroup instead of to the solvable radical as in the Babai–Beals treatment; revisit Figure 1.1 for an illustration.

Along with the proposal of such a model inevitably come questions as to its efficacy. We address two of the more critical issues here. First, our model samples a large number of groups:

Theorem C. *A random $d \times d$ group over \mathbb{Z}/b , as above, samples from each of the following classes of groups.*

(i) *finite abelian groups of exponent dividing b and order at most $O(b^{d^2/2})$.*

(ii) *For each \mathbb{Z}/b -bilinear map $* : U \times V \rightarrow W$, with $\text{rank } U + \text{rank } V + \text{rank } W \leq d$, the Brahana groups [B3] $Bh(*) = U \times V \times W$ with product (also denoted by $*$) as*

$$(u, v, w) * (u', v', w') = (u + u', v + v', w + w' + u * v').$$

(iii) *For each alternating \mathbb{Z}/b -bilinear map $* : U \times U \rightarrow W$, with $\text{rank } U + \text{rank } W \leq d$, the Baer groups [B3] $Br(*) = U \times W$ with product (also denoted by $*$) as*

$$(u, w) * (u', w') = (u + u', w + w' + u * v').$$

(iv) *All classical groups $T(r, q)$ for rank r over \mathbb{F}_q where $r \log q \leq d$.*

(v) *All permutation groups of degree at most d .*

In particular this class of groups samples from $p^{\Theta(d^3)}$ pairwise non-isomorphic groups of order p^d which is a logarithmically dense set of all isomorphism types of groups of order p^d . Furthermore, this class of groups is closed to direct products and subdirect products.

Secondly, for groups selected from our model, even a genus-1, 1-WL refinement results in a filter with constant average width. (Note, constant max width would result in a polynomial-time isomorphism test.)

Theorem D. *For a random group $G \leq U(d, b)$ sampled by our model, one of the following cases occurs on average when d and b are large enough:*

(a) *$O_b(G)$ is abelian; or*

(b) *G has characteristic WL-filter refinement of length $\Theta(\log |G|)$.*

It was predicted in [W3] that most p -groups P had characteristic filters of length $O(\log |P|)$, owing in part to a result of Helleloid–Martin [HM]. However, outside of examples in [W3, M3] there were no large classes of groups where it could be demonstrated that such a filter could be efficiently computed. In a survey of 500,000,000 groups of order 2^{10} conducted by J. Maglione and the fifth author, it was discovered that 96% of groups admitted a filter refinement by algebraic methods, with most stabilizing at $10 = \log_2 1024$ terms. Furthermore, in a sample of 100,000 p -groups having orders between 100 and 3^{70} , most filters refined to a factor of about 10 times the original length. Theorem D offers a theoretical explanation for those experimental results.

1.4 Testing pseudo-isometry of alternating bilinear maps

One base case for which the application of Weisfeiler–Leman is unlikely to go much further is p -groups of class 2 and exponent p . (This special case has long been considered as difficult as the general group isomorphism problem.) As we have seen in Baer’s correspondence [B3] (cf. Theorem C (iii)), when p is odd testing isomorphism of such groups is equivalent to the following problem: given two alternating bilinear maps $\alpha, \beta : U \times U \rightarrow V$, decide whether they are pseudo-isometric, that is, whether they are the same under the natural action of $\text{GL}(U) \times \text{GL}(V)$.

Let $\Lambda(n, q)$ denote the linear space of all $n \times n$ alternating matrices over \mathbb{F}_q , namely the $n \times n$ matrices G such that $v^t G v = 0$ for all $v \in \mathbb{F}^n$. Note, $v \mapsto v^t$ and $G \mapsto G^t$ denotes transposition on vectors and matrices, respectively. An alternating bilinear map $\alpha: U \times U \rightarrow V$ with $U \cong \mathbb{F}_q^n$ and $V \cong \mathbb{F}_q^m$ will be represented by an m -tuple of $n \times n$ such matrices. Testing pseudo-isometry of alternating bilinear maps translates to the following: given two m -tuples of $n \times n$ alternating matrices over \mathbb{F}_q , $\mathbf{G} = (G_1, \dots, G_m)$ and $\mathbf{H} = (H_1, \dots, H_m)$, decide whether there exists $T \in \text{GL}(n, q)$, such that the *linear spans* of $T^t \mathbf{G} T := (T^t G_1 T, \dots, T^t G_m T)$ and \mathbf{H} are the same. For an odd prime p , testing the pseudo-isometry of alternating bilinear maps over \mathbb{F}_p in time $p^{O(n+m)}$ is equivalent to testing isomorphism of p -groups of class 2 and exponent p in time polynomial in group order. Also note that the naïve brute-force algorithm—enumerating all possible $T \in \text{GL}(n, q)$ —takes time $q^{n^2} \cdot \text{poly}(n, \log q)$.

In [LQ] it was shown that when n and m are linearly related, for all but at most $1/q^{\Omega(n)}$ fraction of $\mathbf{G} \in \Lambda(n, q)^m$, there is an algorithm in time $q^{O(n)}$ to test isometry of \mathbf{G} with an arbitrary $\mathbf{H} \in \Lambda(n, q)^m$.¹ The technique used to derive this result merits further comment. It was inspired by, and can be viewed as a linear algebraic analogue of, a classical combinatorial idea from graph isomorphism testing, namely the individualization and refinement technique. More specifically, it follows the use and analysis of this technique by Babai, Erdős, and Selkow, in the first efficient average-case algorithm for graph isomorphism [BES]. By incorporating the genus concept [BMW1] into the individualization and refinement scheme as used in [LQ, BES] we can both extend and improve this result and at the same time greatly simplify the algorithm. Indeed, we have implemented an effective version of this new algorithm in MAGMA [BJP]. We prove:

Theorem E. *Suppose m is larger than some constant. There is an algorithm that, for all but at most $1/q^{\Omega(m)}$ fraction of $\mathbf{G} \in \Lambda(n, q)^m$, tests the pseudo-isometry of \mathbf{G} to an arbitrary m -tuple of alternating matrices \mathbf{H} , in time $q^{O(n+m)}$.*

We briefly outline a simplified version of the algorithm, which is easy to describe and straightforward to implement. A more detailed description can be found in Section 6.1. The simplified version has already captured the essence of the strategy, but it comes with two small drawbacks. First, it does not work over fields of characteristic 2. Secondly, the average-case analysis does not achieve the level stated in Theorem E. Both issues will be remedied in the algorithm presented in Section 6.2, followed by a rigorous average-case analysis.

Assume we are given two m -tuples of $\mathbf{G} = (G_1, \dots, G_m)$ and $\mathbf{H} = (H_1, \dots, H_m)$ from $\Lambda(n, q)^m$ for sufficiently large m and odd q . Let \mathcal{H} be the subspace of $\Lambda(n, q)$ spanned by \mathbf{H} . Take the first c matrices of \mathbf{G} to form a tuple $\mathbf{A} = (G_1, \dots, G_c)$ for some constant $c < m$. Note,

every pseudo-isometry from \mathbf{G} to \mathbf{H} maps \mathbf{A} to a c -tuple \mathbf{B} of matrices in \mathcal{H} .

This simple observation leads to the following algorithm. (We say two c -tuples of alternating matrices \mathbf{A} and \mathbf{B} are *isometric* if there exists an invertible matrix $T \in \text{GL}(n, q)$ such that $T^t \mathbf{A} T = \mathbf{B}$, and the *autometry* group of \mathbf{A} is $\{T \in \text{GL}(n, q) : T^t \mathbf{A} T = \mathbf{A}\}$.) First, check if the autometry group of \mathbf{A} is too large (larger than $q^{\Omega(n)}$). If so, \mathbf{G} does not satisfy our generic condition. Thus, suppose the autometry group is not too large, and enumerate all possible c -tuples \mathbf{B} in \mathcal{H} . Exhaustively check if any of them is isometric to \mathbf{A} , and, in the case of isometry, check if any isometry between \mathbf{A} and \mathbf{B} extends to an pseudo-isometry between \mathbf{G} and \mathbf{H} . The number of isometries between \mathbf{A} and \mathbf{B} is also not too large, because it is equal to the order of the autometry group of \mathbf{A} .

Note that the coset of isometries between \mathbf{A} and \mathbf{B} can be computed in time $\text{poly}(n, c, \log q)$ over fields of characteristic not 2 [BW, IQ]. Enumerating all possible c -tuples in \mathcal{H} incurs a multiplicative

¹The main result in [LQ] is stated in a so-called linear algebraic Erdős–Rényi model. This model is not essentially different from sampling random alternating matrix tuples. See also Remark 6.20 for some details.

cost q^{cm} . Given an isometry between \mathbf{A} and \mathbf{B} , we can check whether \mathbf{G} and \mathbf{H} are pseudo-isometric in $\text{poly}(n, m, \log q)$. Thus, the overall time complexity is bounded above by $q^{cm} \cdot s \cdot \text{poly}(n, m, \log q)$, where s is the order of the autometry group of \mathbf{A} . As we shall prove in Section 6, there is an absolute constant c such that for almost all m -tuple of $n \times n$ alternating matrices \mathbf{G} , the first c matrices have autometry group of order at most $q^{O(n)}$. Thus, the overall time complexity of the aforementioned isometry test is $q^{O(n+m)}$ for almost all \mathbf{G} and arbitrary \mathbf{H} .

Performance. We implemented the above algorithm in MAGMA with some key adjustments (see Section 6.1 for details). The implementation is publicly available on GitHub as part of a comprehensive collection of tools—developed and maintained by the first and last authors and their collaborators—to compute with groups, algebras, and multilinear functions [BMW2].

Absent additional characteristic structure that can be exploited, the traditional approach to deciding pseudo-isometry between alternating bilinear maps $\alpha, \beta: V \times V \rightarrow W$ is as follows. Let $\hat{\alpha}, \hat{\beta}: V \wedge V \rightarrow W$ denote the linear maps induced by α, β . Compute the natural (diagonal) action of $\text{GL}(V)$ on $V \wedge V$, and decide if $\ker \hat{\alpha}$ and $\ker \hat{\beta}$ —each of codimension $\dim W$ in $V \wedge V$ —belong to the same orbit. An alternative version of brute force is to enumerate $\text{GL}(W)$ 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 $\mathbb{F}_3^5 \times \mathbb{F}_3^5 \rightarrow \mathbb{F}_3^4$, 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.

1.5 On groups with genus-2 radicals

There are examples by the fifth author of non-isomorphic p -groups having all proper nontrivial subgroups of a common order isomorphic, and likewise for quotients [W2]. No amount of local invariants will distinguish such groups, so when a WL-refinement style algorithm such as ours encounters such a group it can go no further. Even so, those examples are low genus and thus isomorphism can be decided efficiently by unrelated methods [BMW1]. However, should these groups arise as $O_p(G)$ for a non-nilpotent group G it remains to contend with them as a base case. Combining the code equivalence technique of [BCGQ], the cohomological techniques of [GQ], and results on the automorphism groups of low-genus groups [BMW1], we are able to get a nearly-polynomial running time for testing isomorphism in an important subclass of such groups.

Theorem F. *Let \mathcal{G} be the class of groups G such that $\text{Rad}(G)$ —the largest solvable normal subgroup of G —is a p -group of class 2, exponent $p \neq 2$, such that G acts on $\text{Rad}(G)$ by inner automorphisms. Given groups G_1, G_2 of order n , it can be decided in $\text{poly}(n)$ time if they lie in \mathcal{G} . If so, isomorphism can be decided, and a generating set for $\text{Aut}(G_i)$ found, in time $n^{O(g+\log \log n)}$, where g is the genus of $\text{Rad}(G)$.*

Structure of the paper. After presenting some preliminaries in Section 2, we detail the construction of the colored hypergraphs and prove Theorem A in Section 3. We then explain the combination of filters and composition series isomorphism in Gpl , proving Theorem B in Section 4. The model of random groups, and the effect of the refinement procedure in this model, are the

subject of Section 5, where Theorems C and D are proved. Finally, we provide the average-case algorithm for p -groups of class 2 and exponent p (Theorem E) in Section 6, and the worst-case algorithm for groups with genus-2 radical (Theorem F) in Section 7.

2 Preliminaries

Notation. Let $[m] = \{1, \dots, m\}$ for $m \in \mathbb{N}$. We use $\begin{bmatrix} n \\ d \end{bmatrix}_q$ to denote the Gaussian binomial coefficient with parameters n, d and with base q . Let $M(n \times n', \mathbb{F})$ (resp. $M(n, \mathbb{F})$) be the linear space of all $n \times n'$ (resp. $n \times n$) matrices over \mathbb{F} . The general linear group of degree n over \mathbb{F} is denoted by $GL(n, \mathbb{F})$. When $\mathbb{F} = \mathbb{F}_q$ for some prime power q , we write simply $M(n, q)$ and $GL(n, q)$ in place of $M(n, \mathbb{F}_q)$ and $GL(n, \mathbb{F}_q)$.

Definitions of bilinear maps. Let U, V, W be vector spaces over a field \mathbb{F} . A (\mathbb{F} -)bilinear map is a function $\alpha: U \times V \rightarrow W$ such that

$$\begin{aligned} (\forall u \in U, \forall v, v' \in V, \forall a, b \in \mathbb{F}) \quad & \alpha(u, av + bv') = a\alpha(u, v) + b\alpha(u, v') \\ (\forall u, u' \in U, \forall v \in V, \forall a, b \in \mathbb{F}) \quad & \alpha(au + bu', v) = a\alpha(u, v) + b\alpha(u', v). \end{aligned}$$

If $\beta: U' \times V' \rightarrow W'$ is another \mathbb{F} -bilinear map, we regard β as a function on the same domain and codomain as α by selecting arbitrary linear isomorphisms $U \rightarrow U', V \rightarrow V',$ and $W \rightarrow W'$. We say $\alpha, \beta: U \times V \rightarrow W$ are *isotopic* if there exists $(f, g, h) \in GL(U) \times GL(V) \times GL(W)$ such that $\beta(f(u), g(v)) = h(\alpha(u, v))$ for all $u \in U, v \in V$, and *principally isotopic* if there is an isotopism of the form $(f, g, 1_W)$. If $U = V$, we often require that $f = g$. We say $\alpha, \beta: V \times V \rightarrow W$ are *pseudo-isometric* if there is an isotopism of the form (g, g, h) , and that they are *isometric* if there is a pseudo-isometry of the form $(g, g, 1_W)$. A bilinear map $\alpha: V \times V \rightarrow W$ is *alternating*, if for any $v \in V, \alpha(v, v) = 0$.

Computational models. Suppose, after fixing bases, that $U = \mathbb{F}^\ell, V = \mathbb{F}^n,$ and $W = \mathbb{F}^m,$ which we regard as column spaces. A bilinear map $\alpha: U \times V \rightarrow W$ can be represented as a tuple of matrices $\mathbf{A} = (A_1, \dots, A_m) \in M(\ell \times n, \mathbb{F})^m,$ where

$$(\forall u \in U, v \in V) \quad \alpha(u, v) = (u^t A_1 v, \dots, u^t A_m v)^t.$$

Suppose $\beta: U \times V \rightarrow W$ is represented by $\mathbf{B} = (B_1, \dots, B_m) \in M(\ell \times n, \mathbb{F})^m.$ The concepts of isotopism and principal isotopism then have natural and straightforward interpretations in terms of these matrices. Namely, we say $\mathbf{A}, \mathbf{B} \in M(\ell \times n, \mathbb{F})^m$ are isotopic, if there exist invertible matrices $T \in GL(\ell, \mathbb{F}), S \in GL(n, \mathbb{F})$ and $R \in GL(m, \mathbb{F}),$ such that

$$T^t \mathbf{A} S = (T^t A_1 S, \dots, T^t A_m S) = \left(\sum_{i=1}^m r_{1,i} B_i, \dots, \sum_{i=1}^m r_{m,i} B_i \right) = \mathbf{B}^R,$$

where $r_{i,j}$ denotes the (i, j) -th entry of R for $i, j \in [m].$ We say \mathbf{A} and \mathbf{B} are principal isotopic if they are isotopic with $R = I_m.$

Similarly, an alternating bilinear map $\alpha: V \times V \rightarrow W$ can be represented by a tuple of alternating matrices. Recall that an $n \times n$ matrix G over \mathbb{F} is alternating if for every $v \in \mathbb{F}^n, v^t G v = 0.$ When \mathbb{F} is not of characteristic 2, this is equivalent to the skew-symmetry 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 $\mathbb{F} = \mathbb{F}_q$).

Then *pseudo-isometry* and *isometry* have analogous formulations in terms of alternating matrix tuples.

Given two tuples of alternating matrices $\mathbf{G}, \mathbf{H} \in \Lambda(n, q)^m$, the set of isometries between \mathbf{G} and \mathbf{H} is denoted as

$$\text{Isom}(\mathbf{G}, \mathbf{H}) = \{T \in \text{GL}(n, \mathbb{F}) : T^t \mathbf{G} T = \mathbf{H}\};$$

the group of autometries (or self-isometries) of \mathbf{G} is denoted as $\text{Aut}(\mathbf{G}) = \text{Isom}(\mathbf{G}, \mathbf{G})$. The set of pseudo-isometries between \mathbf{G} and \mathbf{H} is defined as

$$\Psi\text{Isom}(\mathbf{G}, \mathbf{H}) = \{T \in \text{GL}(n, \mathbb{F}) : \exists T' \in \text{GL}(m, q), T^t \mathbf{G} T = H^{T'}\};$$

the group of pseudo-autometries (or self-pseudo-isometries) of \mathbf{G} is denoted as $\Psi\text{Aut}(\mathbf{G}) = \Psi\text{Isom}(\mathbf{G}, \mathbf{G})$. It is straightforward to see that $\text{Isom}(\mathbf{G}, \mathbf{H})$ is a (possibly empty) coset of $\text{Aut}(\mathbf{G})$, and $\Psi\text{Isom}(\mathbf{G}, \mathbf{H})$ is a (possibly empty) coset of $\Psi\text{Aut}(\mathbf{G})$.

Some algorithms for bilinear maps. We note several of the algorithms we cite are described as Las Vegas randomized algorithm in that they depend on factoring polynomials over finite fields. That is known to be deterministic if the characteristic of the field is bounded. In our input model we are given a list of the group elements, so all primes are bounded and so we cite these as deterministic algorithms.

Theorem 2.1. *Let $\alpha, \beta: U \times V \rightarrow W$ be bilinear maps of vector spaces over a finite field \mathbb{F} .*

1. *In time $\text{poly}(\dim U, \dim V, |\mathbb{F}|)$ one can decide if α, β are principally isotopic [BOW, Theorem 3.7].*
2. *If $U = V$ and the characteristic of \mathbb{F} is not 2, in time $\text{poly}(\dim U, |\mathbb{F}|)$ one can decide if α, β are isometric [IQ].*

In each case an affirmative answer is accompanied by a principal isotopism (or isometry).

We also require the following, which follows directly from Theorem 2.1 by enumerating $\text{GL}(W)$.

Theorem 2.2. *Let $\alpha, \beta: U \times V \rightarrow W$ be bilinear maps of vector spaces over a finite field \mathbb{F} .*

1. *In time $\text{poly}(\dim U, \dim V, |W|^{\dim |W|})$ one can decide if α, β are isotopic. [BOW]*
2. *If $U = V$ and the characteristic of \mathbb{F} is not 2, in time $\text{poly}(\dim U, |W|^{\dim |W|})$ one can decide if α, β are pseudo-isometric [IQ].*

The following theorem is the automorphism version of Theorem 2.1. Note that, unlike the case of graph isomorphism, for the problems here there are no known reductions from the isomorphism version to the automorphism version.

Theorem 2.3. *Let $\alpha: U \times V \rightarrow W$ be a bilinear map of vector spaces over a finite field \mathbb{F} .*

1. *In time $\text{poly}(\dim U, \dim V, |\mathbb{F}|)$, one can compute a generating set for the group of principal autotopisms of α [BOW].*
2. *If $U = V$ and the characteristic of \mathbb{F} is not 2, in time $\text{poly}(\dim U, |\mathbb{F}|)$, one can compute a generating set for the group of autometries of α [BW].*

Remark 2.4 (Shuffles). A bilinear map $* : U \times V \rightarrow W$ can be encoded as a 3-dimensional array. Transposing that array allows us to change swap the roles of U, V, W , for example creating a bilinear map $* : V \times U \rightarrow W$ or $* : W^\dagger \times V \rightarrow U^\dagger$, etc. (Here U^\dagger is the dual space of U). This swapping is functorial and therefore isotopisms are permuted accordingly; cf. [BOW]. So while we highlight the situation for principal isotopisms we could indeed specialize any one of the three spaces. We shall assume throughout that when necessary a bilinear map is shuffled.

3 The colored hypergraph algorithm

A high-level description of our algorithm to construct a colored hypergraph associated to a finite group was given in the introduction. We now provide the details; for convenient reference, an outline is given in Algorithm 1 below.

Algorithm 1 Colored Hypergraph

Input: a finite group G , and integers $g, k \geq 1$

Output: a characteristic filter $\phi: \mathbb{N}^d \rightarrow \text{Norm}(G)$ and a colored hypergraph $\mathcal{H}_\chi^{(g,k)}(\phi)$ upon which $\text{Aut}(G)$ acts as color-preserving automorphisms.

- 1: $\phi \leftarrow$ initial characteristic filter for G . Section 1.2
 - 2: Repeat the following steps until ϕ stops changing (stabilizes):
 - a: Build $\mathcal{H}_\chi^{(g)}(\phi)$ on each layer of ϕ . Section 3.1
 - b: Extend $\mathcal{H}_\chi^{(g)}(\phi)$ between layers of ϕ . Section 3.2
 - c: Apply k -dimensional Weisfeiler–Leman to $\mathcal{H}_\chi^{(g)}(\phi)$ Section 3.3
 - d: $S \leftarrow \{\text{Aut}(G)\text{-invariant subgroups extracted from } \text{WL}(k, \mathcal{H}_\chi^{(g)}(\phi))\}$. Section 3.4
 - e: Refine ϕ using S . Section 3.5
 - 3: Return ϕ and $\text{WL}(k, \mathcal{H}_\chi^{(g)}(\phi))$.
-

3.1 Coloring within layers: low-genus quotients and restrictions

For $s \in \mathbb{N}^d$, L_s is a \mathbb{Z}_p -vector space for some prime $p = p_s$ of dimension d_s . Recall that for any vector space L , $\text{PG}(L)$ denotes the *projective geometry* of L , which we may think of as a poset whose elements are the vector subspaces of L , (partially) ordered by inclusion, and we use $\text{PG}_k(L)$ to denote the set of $k+1$ -dimensional subspaces. Let $L_s^* = \text{Hom}(L_s, \mathbb{Z}_p)$ denote the set of linear maps from L_s to \mathbb{Z}_p , i.e., the dual vector space of L_s . Then the map $X \mapsto X^* = \{\nu \in L_s^* : \nu(X) = 0\}$ is an order-reversing bijection $\text{PG}(L_s) \rightarrow \text{PG}(L_s^*)$. By the Fundamental Theorem of Projective Geometry, there is a bijective linear transformation² $f_s: L_s \rightarrow L_s^*$ such that $X^* = f_s(X)$. Let $b_s: L_s \times L_s \rightarrow \mathbb{Z}_p$ be the linear form defined by $b_s(x, y) = f_s(y)(x)$. For $X \leq L_s$, let $X^\perp = \{x \in L_s : b_s(x, X) = 0\}$.

The vertices and hyperedges of $\mathcal{H}^{(g)}(\phi)$ are, respectively,

$$\mathcal{V} = \bigcup_{s \in \mathbb{N}^d} \text{PG}_0(L_s), \quad \mathcal{E} = \left(\bigcup_{s \in \mathbb{N}^d : \dim L_s > g} (\text{PG}_{g-1}(L_s) \cup \text{PG}_{d_s-g-1}(L_s)) \right) \cup \bigcup_{s \in \mathbb{N}^d : \dim L_s \leq g} \text{PG}_{d_s}(L_s). \quad (3.1)$$

(Recall that $L_s \cong \mathbb{Z}_{p_s}^{d_s}$.) To regard $X \in \text{PG}_d(L_s)$ as a hyperedge, when convenient we identify the d -subspace X with the set of points (1-spaces) it contains. The initial coloring is as follows.

- Vertices. The initial color $\chi(v)$ of a vertex $v \in \mathcal{V}$ is simply the index s of the layer L_s such that $v \in \text{PG}_0(L_s)$.

²We note that in some cases, it makes sense to consider a layer L_s as being defined over a larger field \mathbb{F}_{p^k} , thus effectively reducing its dimension, and reducing the size of the hypergraph. In such cases, this map is only guaranteed to be *semi*-linear, that is, $f_s(a+b) = f_s(a) + f_s(b)$, but $f_s(\lambda a) = \alpha(\lambda)f_s(a)$ where $\alpha \in \text{Gal}(\mathbb{F}_{p^k})$ is an automorphism of the field \mathbb{F}_{p^k} . This doesn't present any essential difficulties, but needs to be kept track of.

- Hyperedges corresponding to subspaces of codimension g (dimension $d_s - g - 1$), when $\dim L_s > g$. The initial color $\chi(X)$ of these hyperedges $X \in \text{PG}_{d_s - g - 1}(L_s)$ is determined by s together with a set of labels indexed by pairs $t, u \in \mathbb{N}^d$ such that $t + u = s$ as follows: if $t \neq u$, the label of X corresponding to the pair (t, u) is the isotopism type of the projection $L_t \times L_u \twoheadrightarrow L_s \rightarrow L_s/X^\perp$; when $t = u$ it is the pseudo-isometry type of this projection.
- Hyperedges corresponding to subspaces of dimension g (elements of $\text{PG}_{g-1}(L_s)$), when $\dim L_s > g$. The initial color $\chi(X)$ of these hyperedges is determined by s together with a set of labels indexed by $t \in \mathbb{N}^d$ $t \neq s$ as follows: the label of X corresponding to t is the isotopism type of the restriction of the bimap $L_s \times L_t \twoheadrightarrow L_{s+t}$ to $X \times L_t \twoheadrightarrow L_{s+t}$. (When the dimension is such that dimension g and codimension g subspaces are the same, this set of labels is appended to the set of labels for codimension g subspaces; the two sets of labels are kept separate by their indexing.)
- Hyperedges when $\dim L_s \leq g$. In this case, there is only a single hyperedge X corresponding to the entire layer L_s . It is given a color that is similar to the previous two, namely, for each $t, u \in \mathbb{N}^d$ such that $t + u = s$, $\chi(X)$ gets a set of labels indexed by the pairs (t, u) , labeled by the isotopism type of $L_t \times L_u \twoheadrightarrow L_s$ (resp., pseudo-isometry type if $t = u$), together with, for each $t \in \mathbb{N}^d$ (now including $t = s$) the isotopism (resp., pseudo-isometry) type of the bimap $L_s \times L_t \twoheadrightarrow L_{s+t}$.

Observe, one need not pre-compute all isotopism (resp. pseudo-isometry) types. Instead, one can generate labels on the fly by pairwise comparison. Namely, given a new hyperedge X to label, test for isotopism (or pseudo-isometry) between $L_t \times L_u \twoheadrightarrow L_s/X^\perp$ and all distinctly labelled $L_t \times L_u \twoheadrightarrow L_s/Y^\perp$, introducing a new label for X if necessary.

By Theorem 2.2, isotopism and pseudo-isometry of bilinear maps $U \times V \rightarrow W$ can be decided in time $\text{poly}(\dim U, \dim V, |W|^{\dim W})$, and also (by Remark 2.4) in time $\text{poly}(|U|^{\dim U}, \dim V, \dim W)$. (When $g = 2$, this can be decided very efficiently using the algorithm in [BMW1].) It follows that we can label all hyperedges in time $|G|^{O(g)}$. Note that if the characteristic is 2, then even for maps of the form $L_s \times L_s \twoheadrightarrow L_{s+s}$, we only use the isotopism label instead of pseudo-isometry label, because the results of [IQ] are not yet known to extend to characteristic 2. While this is less refined information, it is still useful.

3.2 Coloring between layers

The colored hypergraph $\mathcal{H}_\chi^{(g)}(\phi)$ described in the previous section already contains much local information from which global characteristic structure may be inferred, extracted, and used. However, we can often elucidate further characteristic structure by examining individual commutator relations *between* the layers. Of the various possible strategies one could try, we propose one that is both elementary and effective.

For each distinct pair $s, t \in \mathbb{N}^d$, add to \mathcal{E} the the following edges. For each $x \in L_s, y \in L_t$ such that $[x, y] = 0$ in L_{s+t} (that is, $[x, y] \in \partial\phi_{s+t}$), we add an edge from x to y . For each x, y which do not commute modulo $\partial\phi_{s+t}$, we add a hyperedge of size 3, connecting $x \in L_s, y \in L_t$, and $[x, y] \in L_{s+t}$. Upon refinement, this allows the vertex colors within each layer to affect the colors in the other layers.

3.3 The Weisfeiler–Leman procedure

Given a vertex-and-hyperedge-colored (hereafter just “colored”) hypergraph $H = (\mathcal{V}, \mathcal{E}, \chi)$, where $\chi: \mathcal{V} \cup \mathcal{E} \rightarrow C$ (C a finite set of colors), we show here how to apply the k -dimensional Weisfeiler–

Leman procedure k -WL, originally developed in the context of graphs independently by Babai–Mathon [B2] and Immerman–Lander [IL] (see [CFI] and [B1] for more detailed history). For the case of $k = 1$ (color refinement) applied to hypergraphs, the same procedure was proposed and studied in the very recent preprint by Böker [B1]. In particular, Böker shows that when we consider a graph as a (2-uniform) hypergraph, this procedure coincides with the usual color refinement procedure on graphs.

Let $\text{WL}(k, H)$ denote the colored hypergraph resulting from applying k -WL to H . The two key properties we will need in our application of this procedure are that: (1) $\text{WL}(k, H)$ can be computed from H in $|H|^{O(k)}$ time, and (2) If H' is another colored hypergraph, then H and H' are isomorphic (as colored hypergraphs) iff $\text{WL}(k, H)$ and $\text{WL}(k, H')$ are isomorphic as colored hypergraphs. (In fact, the set of isomorphisms will be the same: $\text{Iso}(H, H') = \text{Iso}(\text{WL}(k, H), \text{WL}(k, H'))$).

We find it simplest to describe the application of WL to hypergraphs by using instead their “incidence (bipartite) graphs.” We believe this bijection between vertex-and-edge-colored hypergraphs and vertex-colored bipartite graphs is essentially folklore; we include it here for completeness. Given a hypergraph $H = (\mathcal{V}, \mathcal{E})$, its *incidence graph* is the bipartite graph $I(H) = (V_L, V_R, E)$ where $V_L = \mathcal{V}$, $V_R = \mathcal{E}$, $E = \{(v, e) \in \mathcal{V} \times \mathcal{E} : v \in e\}$. It is not hard to see that every bipartite graph arises from a unique hypergraph in this manner, so I is a bijection and I^{-1} is well-defined.

An isomorphism between two vertex-and-edge-colored hypergraphs $H_i = (\mathcal{V}_i, \mathcal{E}_i, \chi_i)$ ($i = 1, 2$) is a bijection $f: \mathcal{V}_1 \rightarrow \mathcal{V}_2$ such that (1) $f(\mathcal{E}_1) = \{f(e) : e \in \mathcal{E}_1\} = \{\{f(v) : v \in e\} : e \in \mathcal{E}_1\} = \mathcal{E}_2$, (2) $\chi_1(v) = \chi_2(f(v))$ for all $v \in \mathcal{V}_1$, and (3) $\chi_1(e) = \chi_2(f(e))$ for all $e \in \mathcal{E}_1$. We say that two vertex-colored bipartite graphs $G_i = (V_{L,i}, V_{R,i}, E_i, \chi_i: V_{L,i} \cup V_{R,i} \rightarrow C)$ ($i = 1, 2$) are isomorphic if there are bijections $f_L: V_{L,1} \rightarrow V_{L,2}$ and $f_R: V_{R,1} \rightarrow V_{R,2}$ such that (1) $f(E_1) = \{(f_L(u), f_R(v)) : (u, v) \in E_1\} = E_2$ and (2) $\chi_1(u) = \chi_2(f_L(u))$ for all $u \in V_{L,1}$ and $\chi_1(f_R(v)) = \chi_2(v)$ for all $v \in V_{R,1}$.

Proposition 3.2 (Folklore). *Given two vertex-and-edge-colored hypergraphs H_1, H_2 , there is a natural bijection between $\text{Iso}(H_1, H_2)$ and $\text{Iso}(I(H_1), I(H_2))$; in particular, H_1 is isomorphic to H_2 iff their vertex-colored bipartite incidence graphs are isomorphic. Furthermore, both I and I^{-1} can be computed in $O(V + E)$ time.³*

Proof sketch. Notation as above. Given $\chi: \mathcal{V} \cup \mathcal{E} \rightarrow C$, a vertex-and-edge coloring on a hypergraph $H = (\mathcal{V}, \mathcal{E})$, we get a coloring on the vertices of $I(H)$, which we also denote by χ by abuse of notation. The coloring on $V(I(H)) = V_L \cup V_R$ is the same as before, since $V_L = \mathcal{V}$ and $V_R = \mathcal{E}$. The inverse is similar. The running time results from the fact that H and $I(H)$ can essentially be described by identical underlying data structures.

We show the natural bijection between $\text{Iso}(H_1, H_2)$ and $\text{Iso}(I(H_1), I(H_2))$. Given an isomorphism $f: \mathcal{V}_1 \rightarrow \mathcal{V}_2$ from H_1 to H_2 , we define an isomorphism \hat{f} from $I(H_1)$ to $I(H_2)$ in the natural way: $\hat{f}(v) = f(v)$ for $v \in V_{L,1} = \mathcal{V}_1$, and for $e \in V_{R,1} = \mathcal{E}_1$ we define $\hat{f}(e) = f(e)$, that is, $\hat{f}(e)$ is the vertex in $V_{R,1} = \mathcal{E}_1$ which corresponds to the hyperedge $\{f(u) : u \in e\}$. To see that \hat{f} is an isomorphism we must check that it preserves incidences and colors. For incidences, we have $(v, e) \in E(I(H_1))$ iff $v \in e$ (thinking of $v \in V_{L,1} = \mathcal{V}_1$ and $e \in V_{R,1} = \mathcal{E}_1$) iff $f(v) \in f(e)$ (since f is an isomorphism of hypergraphs) iff $f(v) = \hat{f}(v) \in \hat{f}(e) = f(e)$, by the definition of \hat{f} . To see that the colors are preserved, for $v \in V_{L,1} = \mathcal{V}_1$, we have, by definition (and abuse of notation), that $\chi(v) = \chi(f(v)) = \chi(\hat{f}(v))$, and for $u \in V_{R,1} = \mathcal{E}_1$ we have $\chi(u) = \chi(f(u)) = \chi(\hat{f}(u))$. The inverse construction of an isomorphism $H_1 \rightarrow H_2$ from an isomorphism $I(H_1) \rightarrow I(H_2)$ is essentially gotten by reading all the preceding equations in reverse. \square

³ $V = |\mathcal{V}|$ for hypergraphs and $|V_L| + |V_R|$ for bipartite graphs; $E = |\mathcal{E}|$ for hypergraphs and $|E|$ for bipartite graphs.

Our k -WL procedure is to apply standard (graph) k -WL to $I(H)$, then applying I^{-1} to get back a refined colored hypergraph.

Finally, we recall the k -WL procedure as applied to a vertex-colored graph. If the graph is bipartite and we want to preserve the bipartition (V_L, V_R) —as in our setting—we assume that the vertices in V_L have distinct colors from those in V_R . Given a vertex-colored graph $G = (V, E, \chi: V \rightarrow C)$, k -WL refinement is the following procedure. Each k -tuple of vertices (v_1, \dots, v_k) is initially assigned a color according to its colored, ordered isomorphism type; that is, two such k -tuples (v_1, \dots, v_k) and (u_1, \dots, u_k) are given the same initial color iff (1) $\chi(v_i) = \chi(u_i)$ for all $i = 1, \dots, k$, (2) $v_i = v_j$ iff $u_i = u_j$ for all $i, j \in [k]$, and (3) $(v_i, v_j) \in E(G)$ iff $(u_i, u_j) \in E(G)$ for all $i, j \in [k]$. Two k -tuples $v = (v_1, \dots, v_k)$ and u are said to be i -neighbors if they are equal except that $v_i \neq u_i$. In each step of the refinement procedure, the coloring is refined as follows: the new color of a tuple v is a k -tuple of multisets, where the i -th multiset is the multiset of colors of all the i -neighbors of v . At each stage, the coloring partitions V^k ; the procedure terminates when this partition doesn't change upon further refinement. Once the coloring on V^k has stabilized, we get a new coloring on $V = V(G)$ by defining $\chi'(v)$ for $v \in V$ to be the color of the diagonal k -tuple $(v, v, \dots, v) \in V^k$. We denote the resulting colored graph by $\text{WL}(k, G)$. From G , $\text{WL}(k, G)$ can be trivially computed in time $O(k^2 n^{2k+1})$; the current best-known running time is still $O(k^2 n^{k+1} \log n)$ [IL, Section 4.9]. For more details on running time, implementation, and the properties of k -WL on graphs, see, e.g., [W1, WL, IL, AFKV, DGR].

3.4 Extracting characteristic structure

Each color class of vertices of $\text{WL}(k, \mathcal{H}_\chi^{(g)}(\phi))$ provides (by lifting from $\phi_s/\partial\phi_s$ to ϕ_s along the natural projection) characteristic subsets of G , but not necessarily characteristic subgroups; it is only the latter which can be used to refine the filter ϕ . To get characteristic subgroups instead, we consider the subgroup generated by all the vertices in a given color class. We now write out this procedure more formally.

Let χ' denote the refined coloring function of $\text{WL}(k, \mathcal{H}_\chi^{(g)}(\phi))$. For each $s \in \mathbb{N}^d$, let χ'_s denote the restriction of χ' to the vertices in $\text{PG}_0(L_s)$. For each color c in the image of χ'_s , let $X_{s,c} = \sum_{x \in \text{PG}_0(L_s): \chi'(x)=c} \langle x \rangle$ be the subgroup of L_s generated by the elements that are colored c . Finally, let $\pi_s: \phi_s \rightarrow \phi_s/\partial\phi_s = L_s$ be the natural projection; we lift $X_{s,c}$ to a characteristic subgroup of ϕ_s (and hence of G) as $\pi^{-1}(X_{s,c})$.

Finally, the set of new characteristic subgroups we consider is

$$S = \{\pi_s^{-1}(X_{s,c}): s \in \mathbb{N}^d, c \in \mathbb{N}\} - \{\phi_s: s \in \mathbb{N}^d\}. \quad (3.3)$$

If $S \neq \emptyset$, its members may be supplied to Theorem 3.4 to refine ϕ , in which case step 3 of Algorithm 1 is repeated. If not, then our colored hypergraph is now stable and Algorithm 1 terminates.

3.5 Refining filters

One filter ϕ refines another filter ψ on the same group if the image of ϕ contains that of ψ (the image is the collection of all subgroups in the filter). If ϕ is a characteristic filter and H is a characteristic subgroup such that $\partial\phi_s \leq H \leq \phi_s$ for some s , then ϕ can be refined to a characteristic filter that includes H . This was first introduced in [W3], and shown to be computable in polynomial time by Maglione [M3]:

Theorem 3.4 ([M3, Theorem 1]). *Let ϕ be a filter on G , and $H \trianglelefteq G$ such that there exists $s \in \mathbb{N}^d$ with $\partial\phi_s < H < \phi_s$. Then a filter refining ϕ and including H can be computed in polynomial time. Furthermore, if ϕ and H are characteristic, then so is the refined filter.*

We proceed sequentially through the characteristic subgroups of S , refining ϕ as we go.

3.6 Proof of Theorem A

For part (i), let $s \in \mathbb{N}^d - \{0\}$. Observe, if Step 3 (c) was omitted from Algorithm 1, then colors would only be assigned to hyperedges on points *in fixed layers*. In that case, moreover, the color of a hyperedge in layer L_s is determined completely by pairs $t, u \in \mathbb{N}^d$ with $t + u = s$; the coloring function χ does not depend at all on layers $v \in \partial\phi_s$. That is to say, if Step 3 (c) is omitted, then $\mathcal{H}_\chi^{(g)}(\phi)$ restricted to N/ϕ_s would be identical to the colored hypergraph based on ϕ truncated at ϕ_s . Step 3 (c) colors edges *between* layers using information from layers ‘lower’ in the filter; this means the restricted hypergraph is a refinement of the hypergraph on the truncated filter.

For part (ii), let G and G' be two finite groups. Suppose we first construct $\mathcal{H}_\chi^{(g)}(\phi)$. Next, we construct $\mathcal{H}_{\chi'}^{(g)}(\phi')$ introducing new color for χ' only when it is new to both colored hypergraphs. Evidently, if $G \cong G'$, then $\mathcal{H}_\chi^{(g)}(\phi)$ and $\mathcal{H}_{\chi'}^{(g)}(\phi')$ are isomorphic with identical color sets.

Finally, we analyze the running time. Computing the Fitting subgroup $O_\infty(G)$ and the initial characteristic filter can be done in $\text{poly}(|G|)$ time, even by naive algorithms (which can be improved significantly when G is given by generating permutations, generating matrices, or black-box generators). Building the hypergraph $\mathcal{H}_\chi^{(g)}(\phi)$ can be done in time linear in the number of hyperedges, which is the number of codimension- g subspaces of each layer L_s , which is $\sim |L_s|^{O(g)}$, and thus in total is at most $|G|^{O(g)}$. The hyperedges can then be colored in $\text{poly}(|G|) \times |G|^{O(g)} = |G|^{O(g)}$ time using the isotopism and isometry algorithms (Theorem 2.2). As with k -WL for graphs, k -WL for hypergraphs can be computed in $|V + E|^{O(k)}$, which in our case is $|G|^{O(gk)}$. Extracting the characteristic subgroups from $\text{WL}(k, \mathcal{H}_\chi^{(g)}(\phi))$ can easily be done in $\text{poly}(|G|)$ time, and refining the filter ϕ can then be done in $\text{poly}(|G|)$ time as well [M3] (reproduced as Theorem 3.4 above). The only remaining question is how many times the main refinement loop can run. Because we only refine when a characteristic subgroup K is found which lies strictly in between some ϕ_s and $\partial\phi_s$, and the indices $|\phi_s : K|$ and $|K : \partial\phi_s|$ are both at least 2, refinement can happen at most $\log_2 |G|$ times. Thus the total running time is $|G|^{O(gk)} \log |G| = |G|^{O(gk)}$. \square

3.7 Incorporating additional invariants

Our algorithm is not particular to the initial characteristic filter we choose. In any given group class, further characteristic subgroups (or subsets, or collections of subgroups) may be available which could be used to refine the filter, either at the beginning, or in each iteration of the main loop of Algorithm 1. We give two examples here without much discussion, just to illustrate the concept, without detracting from the main foci of the paper.

First, it may be the case that some of the bimaps $L_s \times L_t \rightarrow L_{s+t}$ are defined over a field larger than \mathbb{Z}_p , i. e., \mathbb{F}_{p^k} for some $k > 1$. If this is true for sufficiently many of the bimaps, we may be able to treat some layers L_s entirely over \mathbb{F}_{p^k} , thus reducing their dimension by k , and reducing the number of vertices in the corresponding factor of the hypergraph by a factor of k in the exponent (from $p^{k\ell}$ to p^ℓ).

Second, as G acts on N by conjugation, and the layers of ϕ are $\text{Aut}(G)$ -invariant, for each $s \in \mathbb{N}^d$ we can compute a linear representation of G/N on the elementary abelian layer $L_s := \phi_s/\partial\phi_s$. Using standard module machinery—for example, the version of the Meataxe algorithm described in [HR]—in time polynomial in $\log |G|$ each G/N -module may be decomposed first into indecomposable summands, and then into isotypic components. The collection of isotypic components is a characteristic subset of subgroups—namely, they can be permuted amongst themselves by the action

of $\text{Aut}(G)$, but that's it. We can either group these into $\text{Aut}(G)$ -orbits of isotypic components to get characteristic subgroups to refine the filters, or keep the characteristic subset of subgroups and incorporate it into Rosenbaum's composition series isomorphism technique, discussed in Section 4.

3.8 The procedure through an example

We examine the procedure with a toy example as follows. Consider the following alternating matrix tuple in $\Lambda(4, 3)^3$, which was also considered in [BOW].

$$\mathbf{A} = (A_1, A_2, A_3) = \left(\begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \right).$$

We construct a bipartite graph $G_{\mathbf{A}} = (L \cup R, E)$, where $L = \text{PG}_0(\mathbb{F}_3^3)$, and $R = \text{PG}_1(\mathbb{F}_3^3)$, so that for $v \in L$ and $U \in R$, $(v, U) \in E$ if and only if $v \in U$. In particular, note that $|L| = |U| = 13$.

For each $v \in L = \text{PG}_0(\mathbb{F}_3^3)$, we choose a non-zero vector on v as its representative. So

$$L = \{(0, 0, 1), (0, 1, 0), (0, 1, 1), (0, 1, 2), (1, 0, 0), (1, 0, 1), \\ (1, 0, 2), (1, 1, 0), (1, 1, 1), (1, 1, 2), (1, 2, 0), (1, 2, 1), (1, 2, 2)\}.$$

For each $U \in R = \text{PG}_1(\mathbb{F}_3^3)$, since U is a 2-dimensional subspace of \mathbb{F}_3^3 , we choose one defining linear equation u^* , $u \in \mathbb{F}_3^3$, as its representative. So U is also

$$U = \{(0, 0, 1), (0, 1, 0), (0, 1, 1), (0, 1, 2), (1, 0, 0), (1, 0, 1), \\ (1, 0, 2), (1, 1, 0), (1, 1, 1), (1, 1, 2), (1, 2, 0), (1, 2, 1), (1, 2, 2)\}.$$

In this notation, $v = (v_1, v_2, v_3) \in L$ connects to $u = (u_1, u_2, u_3) \in U$, if and only if $v_1u_1 + v_2u_2 + v_3u_3 = 0$.

For $v = (v_1, v_2, v_3)^t \in L$, we define $\mathbf{A}_v = v_1A_1 + v_2A_2 + v_3A_3$ in $\Lambda(4, 3)$. We use $\text{rk}(\mathbf{A}_v)$ to give v the vertex color. Using red for rank 2 and blue for rank 4, we have

$$L = \{(\mathbf{0}, \mathbf{0}, \mathbf{1}), (\mathbf{0}, \mathbf{1}, \mathbf{0}), (\mathbf{0}, \mathbf{1}, \mathbf{1}), (\mathbf{0}, \mathbf{1}, \mathbf{2}), (\mathbf{1}, \mathbf{0}, \mathbf{0}), (\mathbf{1}, \mathbf{0}, \mathbf{1}), \\ (\mathbf{1}, \mathbf{0}, \mathbf{2}), (\mathbf{1}, \mathbf{1}, \mathbf{0}), (\mathbf{1}, \mathbf{1}, \mathbf{1}), (\mathbf{1}, \mathbf{1}, \mathbf{2}), (\mathbf{1}, \mathbf{2}, \mathbf{0}), (\mathbf{1}, \mathbf{2}, \mathbf{1}), (\mathbf{1}, \mathbf{2}, \mathbf{2})\}.$$

The first step of refinement uses the colors on the L side to color the vertices on the U side. For example, $(0, 0, 1)$ on the U side is adjacent to $(1, 0, 0)$, $(0, 1, 0)$, $(1, 1, 0)$, $(1, 2, 0)$. So $(0, 0, 1)$ obtains the color as "3 blues and 1 red", or 3B1R for short. We therefore let blue for 4B, green be 3B1R, and red for 2B2R. We then have

$$U = \{(\mathbf{0}, \mathbf{0}, \mathbf{1}), (\mathbf{0}, \mathbf{1}, \mathbf{0}), (\mathbf{0}, \mathbf{1}, \mathbf{1}), (\mathbf{0}, \mathbf{1}, \mathbf{2}), (\mathbf{1}, \mathbf{0}, \mathbf{0}), (\mathbf{1}, \mathbf{0}, \mathbf{1}), \\ (\mathbf{1}, \mathbf{0}, \mathbf{2}), (\mathbf{1}, \mathbf{1}, \mathbf{0}), (\mathbf{1}, \mathbf{1}, \mathbf{1}), (\mathbf{1}, \mathbf{1}, \mathbf{2}), (\mathbf{1}, \mathbf{2}, \mathbf{0}), (\mathbf{1}, \mathbf{2}, \mathbf{1}), (\mathbf{1}, \mathbf{2}, \mathbf{2})\}.$$

Note that these colors, which comes from genus-1 information, already gives the genus-2 isomorphism types.

The second refinement uses the colors of the U side to recolor the vertices on the L side. For example, $(1, 1, 1)$ on the L side is adjacent to $(1, 1, 1)$, $(1, 2, 0)$, $(1, 0, 2)$, $(0, 1, 2)$ on the U side. So

$(1, 1, 1)$ obtains the color as “2 blues and 2 reds”, or 2B2R for short. We therefore let red for 3R1G, blue for 1R2G1B, and green for 2R2B. We then have

$$L = \{(0, 0, 1), (0, 1, 0), (0, 1, 1), (0, 1, 2), (1, 0, 0), (1, 0, 1), \\ (1, 0, 2), (1, 1, 0), (1, 1, 1), (1, 1, 2), (1, 2, 0), (1, 2, 1), (1, 2, 2)\}.$$

It can be checked that we reach at a stable coloring after this step.

Note that these colors suggest the green points form a characteristic set. This characteristic set would generate the whole group, so it does not yield a non-trivial characteristic subgroup. However, this characteristic set is already interesting, because it does suggest that the Weisfeiler–Leman procedure, or even the naive refinement, gives non-trivial information regarding group elements under the action of automorphisms. We discuss how to take advantage of such characteristic subsets in isomorphism testing in the next section.

4 Isomorphism testing using the filter and hypergraph

Our colored hypergraph and filter constructions can be used to refine the composition-series isomorphism method of Rosenbaum and Wagner [RW], thereby speeding up the resulting isomorphism test. Here, we present an isomorphism algorithm which runs in $\text{poly}(|G|)$ time if the filter output by Algorithm 1 with $k, g \leq O(1)$ also has “width” at most $O(1)$ (defined below). Though we claim no asymptotic improvements in the worst case, we expect our test to perform well for many specific group classes, as well as for groups chosen randomly (including groups selected from the random model we discuss in detail in the following section). In practice, one should also apply Rosenbaum’s bidirectional collision technique [R2] to get a square-root speed-up, but this causes no new technical difficulties.

In fact, the running time we get is $n^{(1/2)\text{width}(\phi_{g,k})+O(1)} + n^{O(gk)}$. We note that the largest that both g and the width can be is $\log n$; if we allow g to be near-maximal (take $kg = \log n / \log \log n$), and this results in a filter whose width is just slightly *less* than maximal, say, $O(\log n / \log \log n)$, then the entire algorithm runs in time $n^{O(\log n / \log \log n)}$, asymptotically beating the trivial algorithm by a $\log \log n$ factor in the exponent. Because this is such a generous bound on kg and a weak desired outcome for the width, we expect this runtime to hold for many classes of groups.

We begin with a simple version, building up to Theorem B in steps.

4.1 Simple version: choose composition series compatible with the filter

We begin by recalling the composition-series isomorphism technique of Rosenbaum and Wagner [RW], and show the simplest way to incorporate our characteristic filter into that technique. (Recall that, although we are not using the colored hypergraph here directly, it contributed to the construction of the filter.) Composition Series Isomorphism is the following problem: given two groups G, H , and a composition series of each $1 \trianglelefteq G_1 \trianglelefteq \cdots \trianglelefteq G_m = G$ and $1 \trianglelefteq H_1 \trianglelefteq \cdots \trianglelefteq H_m = H$, decide whether there is an isomorphism $\varphi: G \rightarrow H$ such that $\varphi(G_i) = H_i$ for all $i = 1, \dots, m$. Rosenbaum and Wagner [RW] show how to reduce p -group isomorphism to Composition Series Isomorphism, and then how to reduce the resulting Composition Series Isomorphism Problem to Graph Isomorphism on graphs of degree at most $p + O(1)$; Rosenbaum more generally showed how to reduce Gpl to Composition Series Isomorphism in $n^{(1/2)\log n + o(\log n)}$ time. Luks [L1] showed how to solve Composition Series Isomorphism in $\text{poly}(n)$ time. Recall the *socle series* of a group G is defined as follows: the *socle* $\text{Soc}(G)$ is the subgroup generated by all minimal normal subgroups. $\text{Soc}(G)$ is always a direct product of simple groups. We then recursively define $\text{Soc}^{i+1}(G)$ to be

the preimage of $\text{Soc}(G/\text{Soc}^i(G))$ in G , that is, if $\pi_i: G \rightarrow G/\text{Soc}^i(G)$ is the natural projection, then $\text{Soc}^{i+1}(G) = \pi_i^{-1}(\text{Soc}(G/\text{Soc}^i(G)))$. The reduction is to pick a composition series for G that is compatible with its socle series, and then to try all possible composition series for H compatible with its socle series. One of the keys to their running time is to show that the number of composition series compatible with the socle series is bounded by $n^{(1/2)\log n}$.

Within $O_\infty(G)$, we refine the socle series with our characteristic filter. Without loss of generality, we may assume that the restriction of our characteristic filter ϕ to the Fitting subgroup $O_\infty(G)$ refines the socle series of $O_\infty(G)$. If it doesn't originally, we may further refine it using the socle series, then iterate the main loop of Algorithm 1 until it stabilizes again. Our algorithm here is to reduce to Composition Series Isomorphism, but to only consider composition series that are compatible both with our filter ϕ and with the socle series. If the filter has many small layers, this will cut down the number of composition series that need to be considered, thus reducing—for such groups—the dominant factor in the running time of [RW, R2, R3].

To illustrate the potential savings, we define the *width* of a filter ϕ with elementary abelian layers to be the maximum dimension of any layer:

$$\text{width}(\phi) := \max_s \dim_{p_s}(\phi_s/\partial\phi_s).$$

Then we have:

Theorem 4.1. *Let N be a solvable group of order n , and ϕ_N be a characteristic filter on N computable in time $t(n)$. Then isomorphism of N with any group can be tested, and an isomorphism found, in time $n^{(1/2)\max_P \text{width}(\phi_P)+O(1)} + t(n)$.*

In particular, using the characteristic filter $\phi_{g,k}$ output by Algorithm 1 with parameters g and k , isomorphism of solvable groups can be solved in time

$$n^{(1/2)\text{width}(\phi_{g,k})+O(1)} + n^{O(kg)}.$$

Proof. The outline of the algorithm follows Rosenbaum–Wagner [RW], also using Luks's polynomial-time algorithm for Composition Series Isomorphism [L1]; the key difference here is that we only consider composition series which refine our characteristic filter ϕ , rather than more general composition series as in Rosenbaum and Wagner. The runtime of their algorithm is a product of the running time to enumerate the desired composition series, and the time to solve Composition Series Isomorphism. Our improvement is in the first step. So we only need calculate the number of composition series of N compatible with $\phi_{g,k}$. In our case, we must first compute the filter ϕ , which takes time $t(n)$.

Let M be a second solvable group. Enumerating the composition series of M compatible with ϕ_M can be achieved as follows. Go through $s \in \mathbb{N}^d$ in lexicographic order, starting with the lexicographically largest s such that $\phi_s \neq 1$. Within each layer $L_s = \phi_s/\partial\phi_s$ we choose all possible composition series. By [RW, Lemma 3.1], this can be done in time $|L_s|^{(1/2)\log_{p_s} |L_s|} \leq |L_s|^{1/2\text{width}(\phi)}$. Taking the product over all layers, we get a bound of $|M|^{(1/2)\text{width}(\phi)}$. For each such composition series, we then use Luks's $\text{poly}(|M|)$ -time algorithm for Composition Series Isomorphism, yielding the stated result.

For the “in particular,” we compute $\phi_{g,k}$ using Algorithm 1, which takes $n^{O(gk)}$ time. \square

4.2 Intermediate version: choose composition series compatible with the filter and hypergraph

The vertex coloring of the hypergraph $\mathcal{H}_\chi^{(g,k)}(\phi)$ may inform us of characteristic subsets that are not subgroups. Although the filter has been refined as much as possible (in particular, any one

of the color classes of the hypergraph in a given layer L_s must generate the whole layer), we can nonetheless take advantage of these characteristic subsets in the preceding algorithm, by further restricting the composition series that we need to consider.

Towards this end, for each layer L_s let C_s denote the smallest color class in L_s , and define the *color ratio* of a layer L_s as $|L_s|/|C_s|$. Finally, define the color ratio of a solvable group N as

$$\text{color-ratio}(N) := \prod_{s \in \mathbb{N}^d} \text{color-ratio}(L_s) = \prod_s \frac{|L_s|}{|C_s|} = \frac{|N|}{\prod_s |C_s|}.$$

We now restate (a slightly refined version) of Theorem B:

Theorem B (Refined). *Let N be a solvable group of order n . Let $\phi = \phi_{g,k}$ and $\mathcal{H}_\chi^{(g,k)}(\phi)$ be the filter and colored hypergraph for N output by Algorithm 1 with parameters g, k . In each layer L_s , let C_s denote the smallest color class. Then isomorphism of N with any group can be tested, and an isomorphism found, in time*

$$\begin{aligned} & \left(\prod_{s \in \mathbb{N}^d} \min\{|L_s|^{1/2}, |C_s|\} \right)^{\text{width}(\phi_{g,k})} \text{poly}(n) + n^{O(gk)} \\ & \leq \left(\frac{n}{\text{color-ratio}(N)} \right)^{\text{width}(\phi_{g,k})} \text{poly}(n) + n^{O(gk)}. \end{aligned}$$

Proof of Theorem B. The outline of the algorithm is the same as in Theorem 4.1; the key difference is how we enumerate composition series within each layer L_s (and how many we enumerate). To see how to take advantage of the size of the smallest color class $C_s \subseteq L_s$, we must recall the details of Rosenbaum & Wagner’s Lemma 3.1 [RW], on enumerating composition series. In the algorithm of Theorem 4.1 above we have already taken care of the ordering of the layers, so the only difference here will be on how we enumerate the part of the composition series within each layer L_s . That is, we may assume that we have already built a composition series of $\partial\phi_s$, which we now want to extend to a composition series of ϕ_s . Since the subgroup generated by C_s would be a characteristic subgroup of L_s , and ϕ has already been refined according to the coloring χ on \mathcal{H} , it must be the case that C_s generates all of L_s . Thus we may select only those composition series where the generator of each step of the composition series comes from C_s . Since any generating set (and hence any composition series) for L_s has size $\log_{p_s} |L_s| \leq \text{width}(\phi)$, the number of choices of composition series where all the generators in the series are chosen from C_s is bounded by

$$|C_s|(|C_s| - 1)(|C_s| - 2) \cdots (|C_s| - \log_{p_s} |L_s| + 1) \leq |C_s|^{\text{width}(\phi)}.$$

This analysis already gives the second bound in the statement of the theorem. To get the more refined bound, within each layer L_s , if $|C_s| < |L_s|^{1/2}$, then we employ the above strategy, and otherwise we use the $|L_s|^{(1/2) \log_{p_s} |L_s|} \leq |L_s|^{(1/2) \text{width}(\phi)}$ strategy from Rosenbaum–Wagner [RW]. \square

4.3 Advanced version: refine the filter and hypergraph as you go (individualize and refine)

Finally, we give a version of the individualize-and-refine paradigm from Graph Isomorphism as applied to composition series that are compatible with our filter and colored hypergraph. The algorithm is similar to that from the previous section, except now, each time we pick a subgroup

in our composition series, we give a new color to the corresponding vertex in our hypergraph, and then we run more iterations of the main loop of Algorithm 1 until the filter and hypergraph again stabilize, before we pick the next subgroup in our composition series. This can potentially have the effect of reducing the width of the layers and/or the size of the smallest color class in each layer as we go.

In somewhat more detail: compute the filter ϕ and colored hypergraph $\mathcal{H}_\chi^{(g,k)}(\phi)$ as before. We build up a composition series in G and simultaneously keep a list of partial composition series in H that we want to test for isomorphism in the end. Suppose we are at the point where we already have built a composition series up to $\partial\phi_s$ in G , and we have a list \mathcal{L} of composition series up to $\partial\phi_s$ in H . Then we extend the partial composition series of G by picking an element of C_s (the smallest color class in L_s). We then color the corresponding vertex in \mathcal{H} a new color, and refine both \mathcal{H} and ϕ until stabilization (as in the main loop of Algorithm 1). Within H , we try each element of C_s in turn, refining the filter and hypergraph for H . If for any $x \in C_s(H)$ the refinement does not agree with the refinement we got in G , we throw it away, otherwise we extend our composition series for H by the subgroup generated by x and $\partial\phi_s$, and add this new partial composition series to our list \mathcal{L} . This comes at a multiplicative cost of $|C_s|$. We then continue this process within the (potentially new, smaller C_s) until we get a composition series that now includes all of ϕ_s . The total multiplicative cost within the layer L_s is thus at most $|C_s|(|C_s| - 1) \cdots (|C_s| - \log_{p_s} |L_s| + 1) \leq |C_s|^{\text{width}(\phi)}$, so this at most squares the total running time from Theorem B.

Thus, asymptotically, we get a similar worst-case upper bound. We could state a more refined upper bound along the lines of Theorem B, but the definitions involved are somewhat delicate and recursive (because they depend on how the width and the color-ratio *change* as the algorithm progresses). Nonetheless, in practice, we expect this individualize-and-refine technique to perform much better, as the layers and color class should decrease in size as the algorithm progresses.

5 A random model of groups

Inspired by a suggestion of A. Mann [M4, Question 8] (answered in [KTW]), we describe here a model for random finite groups. We first give a simplified model that samples only random finite nilpotent groups. Later we extend this to sample solvable, semisimple, and general finite groups.

5.1 A model for random finite nilpotent groups

As a first approximation we choose ℓ random upper unitriangular $(d \times d)$ -matrices u_1, \dots, u_ℓ , over the integers modulo a fixed positive integer b . The u_i are drawn according to a fixed distribution $\mu(d, b)$. Later we shall discuss the effect of μ on the group theory, but first we survey the possible outcomes.

An immediate observation is that $U = \langle u_1, \dots, u_\ell \rangle$ is a subgroup of the full group of upper unitriangular matrices. Therefore, U is nilpotent of order at most $b^{d^2/2}$. In particular, if U_p denotes the Sylow p -subgroup of U , then $U = \prod_{p|b} U_p$. The choice of ℓ generators also has a fingerprint within the structure of our groups U . In particular by Burnside's Basis Theorem, for each $p|b$,

$$|U_p : [U_p, U_p]U_p^p| \leq p^\ell.$$

Thus, there is a certain amount of structure which is fixed by the choice of parameters (d, b, ℓ) . Nevertheless, the coverage asserted in Theorem C shows the diversity of these groups.

5.2 General model

To sample a more general class of groups, we add terms to the block-diagonal. Sampling random invertible square matrices will almost always generate the entire general linear group. As noted in Section 1, a more nuanced approach is called for.

Our strategy is as follows:

- (a) Add solvable groups by selecting any matrix that is diagonalizable over the algebraic closure. We call this a *random toral subgroup*.
- (b) From the classification of finite simple groups we can select at random, according to a fixed distribution, a non-abelian finite simple group T and let $T \leq S/Z(S) \leq \text{Aut}(T)$ —that is, choose S , a (possibly trivial) central extension of an almost simple group. Then we form the group algebra $A = (\mathbb{Z}/b)\langle S \rangle$. We then sample from the minimal left ideals I of A . This defines a linear representation $\rho : T \rightarrow \text{End}(I)$ where I is a \mathbb{Z}/b -module. It is a straightforward exercise to see that endomorphisms of finite modules are representable as chequered matrices. We copy the image of a generating set for S into chequered matrices, and then place this on the block diagonal. We repeat until we exceed a bound on d .
- (c) Add permutation to the block diagonal to any two terms with isomorphic representations.
- (d) As a final step we now sample block upper unitriangular matrices.

It is important to proceed in this order to avoid redundant choices. The number of variability of the simple modules represented on the block diagonal is again controlled by the distribution and that can have substantial impact on the resulting group.

Proposition 5.1. *The class of groups sampled includes: A permutation group P , central extensions S_a of almost simple groups, $P \wr (S_1 \times \cdots \times S_s) \rtimes U$ where U is sampled as above and $L(U)$ is a $\prod_a T_a$ -module.*

5.3 Coverage: Proof of Theorem C

For (i) consider matrices of the form $u_{ij} = I + a_{ij}E_{ij}$. If $1 \leq i < d/2 \leq j \leq d$ then all such u_{ij} commute and are independent. So fix a divisor chain $e_1 | \cdots | e_s | b$ and coefficients a_{ij} (in some index order) having additive order e_{im+j} , it follows that these u_{ij} generate an abelian group with the specified invariants.

For (ii-iii), let $R_v : U \rightarrow W$ where $R_v(u) = u*v$, represented as a matrix. Let \bar{U} be representation of U as in (i), and likewise with \bar{W} . Then

$$\begin{aligned}
 Bh(*) &\cong \left\{ \left[\begin{array}{ccc} 1 & u & w \\ 0 & I_r & R_v \\ 0 & 0 & I_s \end{array} \right] \mid u \in \bar{U}, v \in V, w \in \bar{W} \right\} \\
 Br(*) &\cong \left\{ \left[\begin{array}{ccc} 1 & u & w \\ 0 & I_r & R_u \\ 0 & 0 & I_s \end{array} \right] \mid u \in \bar{U}, w \in \bar{W} \right\}
 \end{aligned}$$

For the count note that it suffices to count the number of distinct bilinear maps $* : U \times V \rightarrow W$. As Higman demonstrates [H2], there are $p^{\dim U \dim V \dim W} / |\text{GL}(U) \times \text{GL}(V) \times \text{GL}(W)| \in p^{\Theta(n^3)}$ such maps.

Finally, for a given list of groups sampled in smaller dimensions, form block diagonal representations. This affords the direct product of the list. For subdirect products take a subgroup of the block-diagonal group. This completes the proof of Theorem C.

5.4 The importance of the right distribution

If we sample dense matrices when we shall call the result the *dense random subgroup model* for the general linear group $\text{GL}(d, \mathbb{F}_p)$. While this is an easy model to reason about it is also fairly rigid, as the following result illustrates.

Theorem 5.2. *If u_1, \dots, u_ℓ are chosen uniformly at random from the group of upper uni-triangular matrices $U_d(\mathbb{Z}/b)$ and $\ell \in \Omega(\sqrt{d})$ then*

$$\Pr\left(|\langle u_1, \dots, u_\ell \rangle| = b^{\ell + \binom{d-2}{2}}\right) \rightarrow 1.$$

In fact we shall prove the following stronger claim: with high probability, such groups $\langle u_1, \dots, u_\ell \rangle$ contain the group of commutators of $U_d(\mathbb{Z}/b)$.

In this model, groups can range widely in isomorphism types, one does not see much variability in coarse isomorphism invariants such as group order, numbers of subgroups or quotients, conjugacy classes, and so forth.

Our proof of Theorem 5.2 relies on some details of Sims' proof on the asymptotic upper bound on the number of isomorphism types of p -groups [S]. It begins as follows. For a group G let $\gamma_i(G)$ be the i th term in the lower central series. Every p -group G has a subgroup $H \leq G$ such that $\gamma_2(H)\gamma_3(G) = \gamma_2(G)$ and where $d(H)$ the least number of group elements to generate H , is minimal with that property [BNV, Proposition 3.8]. We call $d(H)$ the the *Sims' rank* of G .

Definition 5.3. A *Sims subgroup* of a nilpotent group G is a subgroup $H \leq G$ minimal with respect to $\gamma_2(H)\gamma_3(G) = \gamma_2(G)$. The *Sims rank* of G is the minimum number of generators needed to generate a Sims subgroup.

Fix $G = U(d, k)$, $V = G/\gamma_2(G) \cong \mathbb{F}^{d-1}$, $W = \gamma_2(G)/\gamma_3(G) \cong \mathbb{F}^{d-2}$. Then there is a bimap $*$: $V \times V \rightarrow W$ given by commutation:

$$[(\gamma_2(G)x), (\gamma_2(G)y)] \equiv [x, y] \pmod{\gamma_3(G)}. \quad (5.4)$$

Fix a subgroup H of G , and put $U = H\gamma_2(G)/\gamma_2(G)$. Observe that H is a Sims subgroup if, and only if, $[U, U] = [V, V]$. Also observe that after taking natural bases for \mathbb{F}^{d-1} and \mathbb{F}^{d-2} , the bimap $*$ can be represented as follows. Let $[\cdot, \cdot] : \mathbb{F}^{d-1} \times \mathbb{F}^{d-1} \rightarrow \mathbb{F}^{d-2}$ be defined, in a parametrized form, by $[u, v] = uBv^t$ where

$$B = \begin{bmatrix} 0 & f_1 & & & & \\ -f_1 & 0 & -f_2 & & & \\ & -f_2 & \ddots & \ddots & & \\ & & \ddots & \ddots & f_{d-2} & \\ & & & -f_{d-2} & 0 & \end{bmatrix}. \quad (5.5)$$

That is, B could be understood as a 3-tensor of size $(d-1) \times (d-1) \times (d-2)$, whose i th frontal slice is given according to f_i .

Proof of Theorem 5.2. Our approach is to show that a subgroup generated by enough elements is a Sims subgroup. To do this it suffices to show that for most sufficiently large dimensions, the bilinear map of (5.5) has the property that most $X \leq \mathbb{F}^{d-1}$ satisfy $[X, X] = \mathbb{F}^{d-2}$. For notation we let $V = \mathbb{F}^{d-1}$ with basis $\{e_1, \dots, e_{d-1}\}$ and $W = \mathbb{F}^{d-2}$ with basis $\{f_1 = [e_1, e_2], \dots, f_{d-2} = [e_{d-2}, e_{d-1}]\}$.

If $X \leq V$ is the row span of the full rank $(s \times (d-1))$ -matrix M then

$$(MBM^\dagger)_{ij} = \sum_{k=1}^{d-2} (M_{ik}M_{j(k+1)} - M_{i(k+1)}M_{jk})f_k.$$

This defines a natural 3-tensor of size $s \times s \times (d-2)$ by

$$m_{i,j,k} = (M_{ik}M_{j(k+1)} - M_{i(k+1)}M_{jk}).$$

Notice $[X, X] = W$ if, and only if, $\langle \sum_k m_{ijk}f_k | 1 \leq i, j \leq s \rangle = W$. That is, if we flatten the tensor into a $(s^2 \times (d-2))$ -matrix \tilde{m} , as follows,

$$\tilde{m}_{(s \cdot (i-1)+j),k} = (M_{ik}M_{j(k+1)} - M_{i(k+1)}M_{jk}) = \det \begin{bmatrix} M_{ik} & M_{i(k+1)} \\ M_{jk} & M_{j(k+1)} \end{bmatrix};$$

then we are asking that \tilde{m} is of full rank. Now we argue that for $s \geq 2\sqrt{d}$ this is the expected behavior.

By our model, each entry in M is drawn independently at random. However the entries of m (and therefore \tilde{m}) are dependent. Nevertheless, we can observe that the values of m_{ijk} are almost independent of k . Certainly m_{ijk} is independent of $m_{ij'k'}$ if $|k - k'| > 1$. Also, if $k' = k + 1$, if $m_{ijk} \neq 0$ then nothing can be said about $m_{ij(k+1)}$. Even if $m_{ijk} = 0$ it may be impossible to predict $m_{ij(k+1)}$, the exception is when $M_{i(k+1)} = 0 = M_{j(k+1)}$. So there is $1/q^2$ chance of dependence between with the exception of pairs of 0. Each such dependency will be compensated for by adding a row j' such that $M_{j'(k+1)} \neq 0$. Thus $m_{ij'k}$ will be independent of $m_{ij'(k+1)}$. Since the selection of a nonzero entry is a $1 - 1/q \geq 1/2$ event the addition of one row is highly likely to break dependence. Thus, at a cost of sampling $O(\sqrt{d})$ rows we obtain with high probability a matrix M whose associated matrix \tilde{m} is full rank. \square

5.5 Sparsity

For added variation a different distribution is required, one which favors sparse matrices. Fix positive integers b and d . Let $wt(u)$ be the number of non-zero values in the upper unitriangular u . Fix a distribution μ on $\mathbb{Z}/b - \{0\}$ and a distribution ν on $\{1, \dots, \binom{n-1}{2}\}$. Define a (b, d, μ, ν) -random triangular matrix as an $\alpha : \binom{d}{2} \rightarrow \mathbb{Z}/b$ sampled according to a distribution $|\text{supp } \alpha| = k$ with probability $\nu(k)$ and for each $\{i, j\} \in \text{supp } \alpha$, $\alpha_{i,j}$ is sampled according to μ . Notice α uniquely determines an upper unitriangular matrix:

$$u(\alpha) = I_d + \sum_{i=1}^d \sum_{j=i+1}^d \alpha_{ij}E_{ij}. \quad (5.6)$$

The distribution ν describes how large the support of α is expected to be, and μ describes what non-zero values in \mathbb{Z}/b will be used as entries.

Finally define a (μ, ν) -random unitriangular group as the group generated by independently sampling ℓ upper unitriangular $(d \times d)$ -matrices over \mathbb{Z}/b according to their (μ, ν) -distribution. The precise outcomes of this distribution appear intricate. Through some empirical testing (e.g. Figure 1.2) we have produced the following question:

If $\nu(|A|) \rightarrow 0$ for $|A| > C$, does $\log |\langle u_1, \dots, u_\ell \rangle|$ approach a discrete Gaussian distribution on $\{1, \dots, \binom{d}{2}\}$?

Our model makes several constraining choices in order that it avoids the analysis that would otherwise create rather similar groups. The cost of this is that we can so far only offer heuristic explanations for the behavior. Even so, we explain what we understand and encourage a thorough exploration in the future.

The first question is what to expect the length of the block diagonal to be in U . Suppose we assume that the block diagonal is chosen uniformly at a partition of d . From Vershik's theorem [V], the shape of the tableaux of random partition of d with at least \sqrt{d} terms tends to $O(e^{-t})$. That implies that there are relatively few large blocks as those are in the tail of the random distribution. Thus there would be many blocks of small size. This however requires one justify that sampling U at random samples partitions of d uniformly at random. That need not be the case. So we ask

Is the typical sparsely sample group $\langle u_1, \dots, u_\ell \rangle$ convex (tending toward the middle) or concave (tending away from the middle)?

The answer to this speaks to the expected nilpotence class of the groups U . The length of this block diagonal is a bound on the nilpotence class. For example, if there are just two blocks, then

$$U \leq \left\{ \begin{bmatrix} I_a & * \\ 0 & I_b \end{bmatrix} \right\}$$

implies that U is abelian. In general, if \mathcal{F} denotes the subspace flag determining the block structure of U , then the nilpotence class of U is at most $|\mathcal{F}| - 1$.

To see a reason that sparse matrices should sample a wider class of groups than dense matrices we consider a sufficient condition to *avoid* being a Sims subgroup.

Lemma 5.7. *Fix an alternating bimap $[\cdot, \cdot] : V \times V \rightarrow W$ with $W = [V, V]$. Let π_1, \dots, π_{d-2} be a basis of W^* and define $(u, v)_i = \pi_i[u, v]$. For $X \leq V$, if there exists an i such that $(X|X)_i = 0$, then $[X, X] \neq W$.*

Proof. If $(X|X)_i = 0$ then for $u \in W$ with $\pi_i(u) = 1$, $u \notin [X, X]$. □

Now here is the situation. The maps $(\cdot)_i : V \times V \rightarrow K$ are alternating bilinear forms, possibly degenerate. The subspaces $X \leq V$ with $(X|X)_i$ are what are known as *totally isotropic*. The number of maximal totally isotropic subspaces of V is $q^{O(m^2)}$ where $m = \dim V - \dim\{v : (v|V) = 0\}$. Therefore the smaller the radical the much large the number of totally isotropic subspaces there are and therefore the less likely that a subspace X generates W . So as we move towards bimaps for unipotent hulls for flags of fixed length at least 3, then the commutator involved will have quotients to alternating forms with large numbers of totally isotropic subspaces. Thus more subspaces will fail to generate W . As result, fewer subgroups will be Sims subgroups. This however is only a crude guide to the number of Sims subgroups and we encourage an actual analysis with better insights.

5.6 WL-refinement in our random model

So now let us consider the effects of refinement in our random model. Our proof is in two parts. Either our unipotent groups U have long block diagonal series or it has bounded class. In the former case we reduce the refinement analysis to a result of Maglione [M3]. In the later case we appeal to classical results on nonsingular products. In either case we discover refinements. We aim to prove Theorem D.

Refinements for many blocks. First let us consider groups with many blocks.

Theorem 5.8. *The refinement length of a random subgroup $U \leq U(d, p)$ is on average at $\Omega(\ell^2)$ where ℓ is the length of its generalized eigen 1-space flag.*

Primarily we want to appeal to the following. Note that in this case we do not apply the Weisfeiler–Leman procedure developed in this paper; instead, it will be used in the next setting.

Theorem 5.9 (Maglione [M2]). *The group $U(d, p)$ has an (adjoint) characteristic filter refinement of length $\Theta(d^2)$.*

However we do not have the group $U(d, p)$. Instead, we have a subgroup sampled at random either with dense or sparse matrices. First we dispense with the dense case.

Corollary 5.10. *A subgroup $H \leq U(d, p)$ generated by dense matrices u_1, \dots, u_ℓ with $\ell \geq 2\sqrt{d}$ has on average a characteristic filter refinement of length $\Theta(d^2)$.*

Proof. By Theorem 5.2, H is almost certainly a Sims subgroups of $U(d, p)$ and therefore $[H, H] = [U(d, p), U(d, p)]$. As a scholium to Maglione’s theorem we observe that the adjoint filter refinement of $U(d, p)$ can be defined as refinement through terms L_s for $s < (2, 0, \dots, 0)$ in the filter. As a result these same terms appear in the filter of H and so H refines to a length of $\Omega(d^2 - d) = \Omega(d^2)$. Since $\log_p |H| \in O(d^2)$ the result follows. \square

Next we need to consider the sparse case as this is where our model presides. What we do is demonstrate a form of *Morita condensation theory* that transports our sparse problem into a dense problem [W4]. What we observe is that each right-hand edge j of block on the block diagonal of U is defined by the presence of an element $u \in U$ with a non-zero value u_{ij} , otherwise the block would be wider. We select one such row i_s for each block s , and one such column j_s . Thus out of the $(\ell \times \ell)$ -block matrix $u \in U$, we create an $(\ell \times \ell)$ -matrix by copying the entire in $u_{i_s j_s}$. For example

$$\left[\begin{array}{cc|ccc} 1 & 0 & a_{13} & a_{14} & a_{15} & a_{16} \\ 0 & 1 & a_{23} & a_{24} & a_{25} & a_{26} \\ \hline & & 1 & a_{34} & a_{35} & a_{36} \\ \hline & & & 1 & 0 & 0 \\ & & & 0 & 1 & 0 \\ & & & 0 & 0 & 1 \end{array} \right] \mapsto \begin{bmatrix} 1 & a_{13} & a_{15} \\ & 1 & a_{35} \\ & & 1 \end{bmatrix}.$$

This may seem a bit unnatural but in fact it is applying a functorial property not on the level of groups but on the level of the enveloping algebra of the matrices and more importantly on the level of bilinear maps. While this function has no relationship in the context of groups, it is by considering the associated ring context that we see that we have simply performed a condensation of modules, that is we have changed to an equivalent category. So for each of the blocks B_1, \dots, B_d we let e_s be the $(d_s \times d_s)$ matrix with zero in every position except jj . Set $e = e_1 \oplus \dots \oplus e_\ell$. Then eue is matrix with at most $\ell \times \ell$ nonzero entries. Removing the all zero rows and columns produces an $(\ell \times \ell)$ -matrix. In the example above,

$$e = \left[\begin{array}{cc|ccc} 1 & 0 & & & \\ 0 & 0 & & & \\ \hline & & 1 & & \\ \hline & & & 0 & 0 & 0 \\ & & & 0 & 1 & 0 \\ & & & 0 & 0 & 0 \end{array} \right]$$

In particular this induces a functorial Morita condensation of each bilinear map $L_s \times L_t \rightsquigarrow L_{s+t}$, see [W4]. We therefore denote this group \overline{eUe} to remind us of the natural process to create this smaller matrix group.

Having applied this transform, notice \overline{eUe} is now a dense subgroup of $U(\ell, p)$. Therefore we arrive at the following.

Proof of Theorem 5.8. Suppose $U \leq U(d, p)$ generate by random matrices u_1, \dots, u_t . If the u_i are dense then by Corollary 5.10 there is a computable filter refinement of length $O(\ell^2)$ where ℓ is the number of blocks of U . If on the other hand the u_i are sparse, then \overline{eUe} has a refinement of length $\Omega(\ell^2)$. As the map $U \mapsto \overline{eUe}$ is functorial in the bilinear maps used to select refinement, it follows that U also has a refinement of length $\Omega(\ell^2)$. \square

Refinements for few blocks. The last case to concern us is when U has a bounded number of blocks on the diagonal, but that the number of blocks is at least 3. (Otherwise the group U is abelian which is the first case of Theorem D.) Because the number of blocks is bounded at least one block has dimension proportional to d as $d \rightarrow \infty$.

Let us consider coloring with $g = 1$. This means that with a selected layer $L_s \times L_t \rightsquigarrow L_{s+t}$ we consider labels on 1-dimensional subspaces $\langle x \rangle \leq L_s$ by labeling the restriction $\langle x \rangle \times L_t \rightsquigarrow L_{s+t}$. One observes this structure is nothing more than a linear transformation $L_t \rightarrow L_{s+t}$ and is thus defined up to change of basis solely by the rank of the transformation. Therefore to each element of $\text{PG}_0(L_s)$ we record the rank of the associated matrix. We do likewise with $\text{PG}(L_t)$. Finally we label the edges between $\text{PG}(L_s)$ and $\text{PG}(L_t)$ by whether or not the pair of points commutes.

In order to model this behavior in colors we make the following observation. Treating $x = (x_1 : \dots : x_d)$ as homogeneous point in d variables, the evaluation $[x, -] : L_t \rightarrow L_{s+t}$ produces a matrix $M(x)$ with entries in $\mathbb{F}[x_1, \dots, x_d]$. The rank of this matrix changes as we evaluate x but certainly there are two natural states: either $M(x)$ has rank at most r or it does not. If $M(x)$ has rank at most r then all $(r \times r)$ -minors must vanish, and this produces a polynomial number degree r -polynomials that must all vanish on x . That is to say, the condition of the rank of $M(x)$ is a variety (or more generally a scheme). It is in fact a *determinantal variety* and the subject of considerable study in the algebraic community as well as the computer science community [H1, FSEDS]. It is important to observe that many results in the field are only known over algebraically closed fields. However it is known that these varieties are reduced and irreducible [H1]. Therefore to count points we can use Lang–Weil theorem [LW1], but that requires that we allow for a large field. So this portion of our estimate assume $b \rightarrow \infty$ and $d \rightarrow \infty$.

Let us assume for now that $M(x)$ has points, i.e. that for some $x \in \text{PG}_0(L_s)$, $[x, -]$ does not have full rank, and for other points it does. Thus our vertex set has (at least) 2 colors, say white if $[x, -]$ has full rank and black otherwise. We do the same for $\text{PG}(L_t)$. Recall that we are including a hyperedge $(x, y, [x, y])$ only if $[x, y] \neq 0$.

Now consider the situation. The number of black points is in general a solution to a system of random nonlinear homogeneous polynomials of degree r . That this is nonlinear means we can expect that the number of black points is not a subspace. Now the points in $\text{PG}(L_t)$ not connected to black points x are the points $y \in x^\perp := \ker[x, -]$. In particular we have a nonlinear set parameterizing a subspace arrangement within $\text{PG}(L_t)$. If we write the generator matrix of each subspace $\ker[x, -]$ it will be the dual of a linear combination of the matrices used to define $[,]$, which we sampled at random. Therefore we have a random subspace arrangement. In general this incidence relation is not equitable, so proper refinements will be discovered in the WL-refinement process.

With that we have proved the following.

Theorem 5.11. *If $U \leq U(d, p)$ has a bounded number of blocks and d, p are large, then there exists a proper refinement of the standard filter.*

To remove the assumption that d, p are large here, the following interesting question needs to be addressed.

Let A_1, \dots, A_m be random $n \times n$ matrices over \mathbb{F}_q . What is the typical number of non-full-rank matrices in the linear span of A_i 's?

5.7 Proof of Theorem D

Let us suppose G is sampled according to our model. Let U be the intersection of G with $U(d, p)$ and begin with the initial filter of our introduction. Then if U is abelian we are in case (i). Otherwise U has at least 3 blocks so we can use either Theorem 5.8 for the case of large blocks, or Theorem 5.11 in the case of small blocks. In either case we obtain a proper refinement. Note that after refinement of the bounded number of blocks several times we cross over to the large number of blocks and so the result follows. \square

6 Testing pseudo-isometry of alternating bilinear maps

6.1 The simplified main algorithm

In this subsection we formally describe the simplified main algorithm presented in Section 1.4, that is Algorithm 2. We also discuss some important adjustments used in the implementation. We need the following observation, which follows easily by computing the closure of the given generating set.

Observation 6.1. *Let $C_1, \dots, C_t \in \text{GL}(n, q)$, 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 $\text{poly}(s, n, \log q)$.*

Let us first examine the running time of Algorithm 2.

Proposition 6.2. *Algorithm 2 runs in time $\text{poly}(q^{cm}, s, n)$.*

Proof. If Algorithm 2 outputs $|\text{Aut}(\mathbf{A})| > s$, then its running time is determined by Theorem 2.3 (2) and Observation 6.1, which together require $\text{poly}(s, n, \log q)$.

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

We then prove the correctness of Algorithm 2, in the case that it does not report $|\text{Aut}(\mathbf{A})| > s$.

Proposition 6.3. *If Algorithm 2 does not report $|\text{Aut}(\mathbf{A})| > s$, then it lists the set of pseudo-isometries (possibly empty) between \mathbf{G} and \mathbf{H} . In particular, $|\Psi\text{Isom}(\mathbf{G}, \mathbf{H})| \leq q^{cm} \cdot s$.*

Proof. By Step 4.c, 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 the linear spans of $T^t \mathbf{G} T$ and \mathbf{H} are the same, we know $T^t \mathbf{A} T$ is equal to some $\mathbf{B} \in \mathcal{H}^c$. So when enumerating this \mathbf{B} in Step 4, T will pass all the tests in the following, and then be added to L . This concludes the proof. \square

Algorithm 2 The first average-case algorithm for alternating space isometry.

Input: $\mathbf{G} = (G_1, \dots, G_m) \in \Lambda(n, q)^m$, $\mathbf{H} = (H_1, \dots, H_m) \in \Lambda(n, q)^m$, $c, s \in \mathbb{N}$, and q is odd.

Output: Either (1) $|\text{Aut}(\mathbf{A})| > s$, where $\mathbf{A} = (G_1, \dots, G_c)$, or (2) $\Psi\text{Isom}(\mathbf{G}, \mathbf{H})$.

Algorithm procedure:

1. Set $L \leftarrow \{\}$. Set $\mathbf{A} = (G_1, \dots, G_c)$, the first c matrices from \mathbf{G} .
 2. Use Theorem 2.3 (2) to compute a generating set for $\text{Aut}(\mathbf{A})$.
 3. Use Observation 6.1 with input s and the generating set of $\text{Aut}(\mathbf{A})$.
(If $|\text{Aut}(\mathbf{A})| > s$, we terminate the algorithm and report that “ $|\text{Aut}(\mathbf{A})| > s$.”)
 4. Put $\mathcal{H} = \langle \mathbf{H} \rangle$, the linear span of \mathbf{H} ; for every $\mathbf{B} = (B_1, \dots, B_c) \in \mathcal{H}^c$, do the following.
 - a. Use Theorem 2.1 (2) to decide whether \mathbf{A} and \mathbf{B} are isometric.
 - b. If not, go to the next \mathbf{B} . Otherwise, we get the non-empty coset $\text{Isom}(\mathbf{A}, \mathbf{B})$.
 - c. For every $T \in \text{Isom}(\mathbf{A}, \mathbf{B})$, do the following.
Test whether the linear spans of $T^t \mathbf{G} T$ and \mathbf{H} are the same. If not, go to the next T . If so, add T into L .
 5. Output L .
-

It remains to specify the choices of c and s in Algorithm 2 in the average-case analysis. This is stated in the following, whose proof will be deferred to Section 6.3.

Proposition 6.4. *Let $c := 20$. For all but at most $1/q^{\Omega(n)}$ fraction of $\mathbf{A} = (G_1, \dots, G_c) \in \Lambda(n, q)^c$, we have $|\text{Aut}(\mathbf{A})| \leq s := q^n$.*

Combining Propositions 6.2, 6.3 and 6.4, we have the following theorem.

Theorem 6.5. *Let $m \geq 20$, and let \mathbb{F}_q be a finite field of odd size. For all but at most $1/q^{\Omega(n)}$ fraction of $\mathbf{G} = (G_1, \dots, G_m) \in \Lambda(n, q)^m$, Algorithm 2 tests the isometry of \mathbf{G} with an arbitrary $\mathbf{H} \in \Lambda(n, q)^m$ in time $q^{O(n+m)}$.*

Implementation details. We now explain some issues in the implementation of Algorithm 2.

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 $\text{Aut}(\mathbf{A})$. 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 G_1, \dots, G_c are low-rank matrices, then we will only need to match them with the low-rank matrices from \mathcal{H} . Our experiment shows that, for a random \mathcal{G} over \mathbb{F}_q when q is a small constant, the number of low-rank (i.e. non-full-rank) matrices in \mathcal{G} 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. So we can use $q^m \cdot \text{poly}(n, \log q)$ to choose 3 low-rank matrices from \mathcal{G} . Then

use $q^m \cdot \text{poly}(n, \log q)$ to compute the set of low-rank matrices from \mathcal{H} , denoted as \mathcal{H}_c . We can then replace enumerating \mathcal{H}^c with \mathcal{H}_l^c , which in general is much smaller.

6.2 The main algorithm

To state our algorithm, we need the concept of adjoint algebra. For two tuples of alternating matrices $\mathbf{G}, \mathbf{H} \in \Lambda(n, \mathbb{F})^m$, the adjoint algebra of \mathbf{G} is defined as

$$\text{Adj}(\mathbf{G}) = \{(A, D) \in \text{M}(n, \mathbb{F}) \oplus \text{M}(n, \mathbb{F}) : \mathbf{A}\mathbf{G} = \mathbf{G}\mathbf{D}\},$$

and the adjoint space from \mathbf{G} to \mathbf{H} is

$$\text{Adj}(\mathbf{G}, \mathbf{H}) = \{(A, D) \in \text{M}(n, \mathbb{F}) \oplus \text{M}(n, \mathbb{F}) : \mathbf{A}\mathbf{G} = \mathbf{H}\mathbf{D}\}.$$

Clearly, if $T \in \text{Aut}(\mathbf{G})$, then $(T^t, T^{-1}) \in \text{Adj}(\mathbf{G})$. Furthermore, if \mathbf{G} and \mathbf{H} are isometric, then $|\text{Adj}(\mathbf{G}, \mathbf{H})| = |\text{Adj}(\mathbf{G})|$.

We now introduce the algorithm (see Algorithm 3) that supports Theorem E. We point out that Algorithm 3 differs from the algorithm presented in Section 1.4 in two places.

1. The first and major difference is to replace the uses of $\text{Aut}(\mathbf{G})$ and $\text{Isom}(\mathbf{G}, \mathbf{H})$ with $\text{Adj}(\mathbf{G})$ and $\text{Adj}(\mathbf{G}, \mathbf{H})$, thereby avoiding using Theorem 2.1 (2) and Theorem 2.3 (2). Since $\text{Adj}(\mathbf{G})$ and $\text{Adj}(\mathbf{G}, \mathbf{H})$ are easy to compute over any field, this resolves the characteristic-2 field issue. Furthermore, $\text{Adj}(\mathbf{G})$ and $\text{Adj}(\mathbf{G}, \mathbf{H})$ are also easier to analyze. But $\text{Adj}(\mathbf{G})$ and $\text{Adj}(\mathbf{G}, \mathbf{H})$ could be larger than $\text{Aut}(\mathbf{G})$ and $\text{Isom}(\mathbf{G}, \mathbf{H})$, so they are less useful from the practical viewpoint.
2. The second place is step 2 in Algorithm 3: instead of just using the first c matrices as in the algorithm presented in Section 1.4, Algorithm 3 slices the m matrices of \mathbf{G} 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 Section 1.4 as well.

Let us first examine the running time of Algorithm 3.

Proposition 6.6. *Algorithm 3 runs in time $\text{poly}(q^{cm}, s, n)$.*

Proof. If Algorithm 3 outputs “ \mathbf{G} does not satisfy the generic condition,” then it just executes the For-loop in Step 3, which together runs in time $\text{poly}(m, n, \log q)$.

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

We then prove the correctness of Algorithm 3 in the case that that it does not report “ \mathbf{G} does not satisfy the generic condition.”

Proposition 6.7. *Suppose that Algorithm 3 does not report “ \mathbf{G} does not satisfy the generic condition.” Then the algorithm lists the set of pseudo-isometries (possibly empty). In particular, $|\Psi\text{Isom}(\mathbf{G}, \mathbf{H})| \leq q^{cm} \cdot s$.*

Proof. By Step 5.c, every 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} to some $\mathbf{B} \in \mathcal{H}^c$, i.e., $T^t \mathbf{A} T = \mathbf{B}$. In particular, $(T^t, T^{-1}) \in \text{Adj}(\mathbf{A}, \mathbf{B})$. So when enumerating this $\mathbf{B} \in \mathcal{H}^c$, (T^t, T^{-1}) will pass all the tests in the following, and then be added to L . This concludes the proof. \square

Algorithm 3 The second average-case algorithm for alternating space isometry.

Input: $\mathbf{G} = (G_1, \dots, G_m) \in \Lambda(n, q)^m$, $\mathbf{H} = (H_1, \dots, H_m) \in \Lambda(n, q)^m$ and $c, s \in \mathbb{N}$.

Output: Either (1) $|\text{Aut}(\mathbf{A})| > s$; or (2) $\Psi\text{Isom}(\mathbf{G}, \mathbf{H})$ as a set, which may be empty.

Algorithm procedure: 1. Set $L \leftarrow \{\}$. Set $F \leftarrow \text{false}$.

2. For $i = 1, \dots, \lfloor m/c \rfloor$, do the following.
 - a. Set $\mathbf{A} = (G_{c(i-1)+1}, \dots, G_{ci})$.
 - b. Compute a linear basis of $\text{Adj}(\mathbf{A}) \subseteq \text{M}(n, q) \oplus \text{M}(n, q)$.
 - c. If $|\text{Adj}(\mathbf{A})| \leq s$, set F to be **true**, and break the For-loop.
 3. If $F = \text{false}$, return “ \mathbf{G} does not satisfy the generic condition.” and terminate. Otherwise,
 4. Put $\mathcal{H} = \langle \mathbf{H} \rangle$, the linear span of \mathbf{H} ; for every $\mathbf{B} = (B_1, \dots, B_c) \in \mathcal{H}^c$, do the following.
 - a. Compute a linear basis for $\text{Adj}(\mathbf{A}, \mathbf{B}) \subseteq \text{M}(n, q) \oplus \text{M}(n, q)$.
 - b. If $|\text{Adj}(\mathbf{A}, \mathbf{B})| > s$, go to the next \mathbf{B} .
 - c. For every $(T, S) \in \text{Adj}(\mathbf{A}, \mathbf{B})$, do the following.

If S and T are invertible and $S = T^{-t}$, test whether the linear spans of $T\mathbf{G}T^t$ and \mathbf{H} are the same. If not, go to the next (T, S) . If so, add T^t into L .
 5. Output L .
-

Therefore, to prove Theorem E, the key is to analyze when a random \mathbf{G} satisfies the generic condition as in Algorithm 3.

Proposition 6.8. *Let $m \geq c = 20$, and let $\ell = \lfloor m/20 \rfloor \in \mathbb{N}$. For all but at most $1/q^{\Omega(n \cdot \ell)} = 1/q^{\Omega(nm)}$ fraction of $\mathbf{G} = (G_1, \dots, G_m) \in \Lambda(n, q)^m$, there exists some $i \in [\ell]$, such that, letting $\mathbf{A} = (G_{c(i-1)+1}, \dots, G_{c(i-1)})$, we have $|\text{Adj}(\mathbf{A})| \leq q^n$.*

Clearly, Theorem E follows from Propositions 6.6, 6.7, and 6.8.

6.3 The average-case analysis

We now formulate the key proposition that supports the proof of Proposition 6.8.

Proposition 6.9. *Let $c = 20$. For all but at most $1/q^{\Omega(n)}$ fraction of $\mathbf{A} = (G_1, \dots, G_c) \in \Lambda(n, q)^c$, we have $|\text{Adj}(\mathbf{A})| \leq q^n$.*

Given Proposition 6.9, we easily obtain the following.

Proof of Proposition 6.4. This is because, if $T \in \text{Aut}(\mathbf{A})$, then $(T^t, T^{-1}) \in \text{Adj}(\mathbf{A})$. So $|\text{Aut}(\mathbf{A})| \leq |\text{Adj}(\mathbf{A})|$. \square

Proof of Proposition 6.8. We slice \mathbf{G} into $\ell = \lfloor m/c \rfloor$ segments, where each segment consists of c random alternating matrices. Each segment is some $\mathbf{A} \in \Lambda(n, q)^c$, with $\text{Pr}[|\text{Adj}(\mathbf{A})| > q^n] \leq 1/q^{\Omega(n)}$. Since each G_i is chosen independently and uniformly at random, the probability of every $(G_{c(i-1)+1}, \dots, G_{ci})$, $i \in [\ell]$, with $|\text{Adj}((G_{c(i-1)+1}, \dots, G_{c(i-1)+c}))| > q^n$, is upper bounded by $(1/q^{\Omega(n)})^\ell = 1/q^{\Omega(nm)}$. \square

The rest of this subsection is devoted to the proof of Proposition 6.9. For this we need the following from [LQ]. Given a tuple $\mathbf{A} = (A_1, \dots, A_r) \in M(n, q)^r$, define the image of $U \leq \mathbb{F}_q^n$ under \mathbf{A} as $\mathbf{A}(U) := \langle \cup_{i=1}^r A_i(U) \rangle$.

Definition 6.10. We say $\mathbf{A} = (A_1, \dots, A_r) \in M(n, q)^r$ is *stable*, if for any nonzero, proper $U \leq \mathbb{F}_q^n$, we have $\dim(\mathbf{A}(U)) > \dim(U)$.

Proposition 6.11 ([LQ, Proposition 10 in arXiv version]). *If $\mathbf{A} \in M(n, q)^r$ is stable, then $|\text{Adj}(\mathbf{A})| \leq q^n$.*

A key technical result in [LQ] is that, a random $\mathbf{A} \in M(n, q)^4$ is stable with probability $1 - \frac{1}{q^{\Omega(n)}}$ [LQ, Proposition 20 in arXiv version]. However, we cannot directly apply that result to prove Proposition 6.9, because here we have alternating matrices instead of general matrices. So we have to run the arguments for the proof of [LQ, Proposition 20 in arXiv version] again, and carefully adjust some of the details there to accommodate the structure of alternating matrices.

To start with, we need the following easy linear algebraic result, which suggests the connection between random alternating matrices and random general matrices.

Lemma 6.12. *Let $d \in \mathbb{Z}^+$ and $d \geq 2$. Given two random alternating matrix $X, Y \in \Lambda(d, q)$, we can construct a matrix $P \in M(d \times (d-1), q)$, whose columns are linear combinations of the columns of X and Y , such that P is a random matrix from $M(d \times (d-1), q)$.*

Proof. Let X and Y be given as

$$X = \begin{bmatrix} 0 & x_{1,2} & x_{1,3} & \dots & x_{1,d} \\ -x_{1,2} & 0 & x_{2,3} & \dots & x_{2,d} \\ -x_{1,3} & -x_{2,3} & 0 & \dots & x_{3,d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -x_{1,d} & -x_{2,d} & -x_{3,d} & \dots & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & y_{1,2} & y_{1,3} & \dots & y_{1,d} \\ -y_{1,2} & 0 & y_{2,3} & \dots & y_{2,d} \\ -y_{1,3} & -y_{2,3} & 0 & \dots & y_{3,d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -y_{1,d} & -y_{2,d} & -y_{3,d} & \dots & 0 \end{bmatrix},$$

where each $x_{i,j}$ and $y_{i,j}$ are independent random variables from \mathbb{F}_q . Define

$$M = \begin{bmatrix} y_{1,2} & x_{1,2} + y_{1,3} & x_{1,3} + y_{1,4} & \dots & x_{1,d-1} + y_{1,d} & x_{1,d} \\ -x_{1,2} & y_{2,3} & x_{2,3} + y_{2,4} & \dots & x_{2,d-1} + y_{2,d} & x_{2,d} - y_{1,2} \\ -x_{1,3} - y_{2,3} & -x_{2,3} & y_{3,4} & \dots & x_{3,d-1} + y_{3,d} & x_{3,d} - y_{1,3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -x_{1,d} - y_{2,d} & -x_{2,d} - y_{3,d} & -x_{3,d} - y_{4,d} & \dots & -x_{d,d-1} & -y_{1,d} \end{bmatrix}$$

$$:= \begin{bmatrix} z_{1,1} & z_{1,2} & z_{1,3} & \dots & z_{1,d-1} & z_{1,d} \\ z_{2,1} & z_{2,2} & z_{2,3} & \dots & z_{2,d-1} & z_{2,d} \\ z_{3,1} & z_{3,2} & z_{3,3} & \dots & z_{3,d-1} & z_{3,d} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ z_{d,1} & z_{d,2} & z_{d,3} & \dots & z_{d,d-1} & z_{d,d} \end{bmatrix},$$

be the matrix obtained by adding the $(i+1)$ th column of Y to the i th column of X for $i \in [d-1]$, and add the first column of Y to the d th column of X . Let P be the $d \times (d-1)$ matrix consisting of the first $(d-1)$ columns of M . We need to show that P is uniformly sampled from $M(n \times (d-1), q)$ as X and Y are uniformly sampled from $\Lambda(d, q)$.

To see this, first note that for any two random variable x and y , which are chosen independently and uniformly at random from \mathbb{F}_q^d , $x \pm y$ are also new random variables which are chosen uniformly

at random from \mathbb{F}_q^d , and is independent with either x or y . Thus each $z_{i,j}$ is again a random variable which is chosen uniformly at random from \mathbb{F}_q^d for $i, j \in [d]$.

We then exploit the linear relations among the $z_{i,j}$'s. In fact, we only need to focus on the anti-diagonal directions, as

$$z_{1,i} + z_{2,i-1} + \cdots + z_{i,1} + z_{i+1,d} + z_{i+2,d-1} + \cdots + z_{d,i+1} = 0$$

for any $i \in [d]$. Thus, we can view $z_{1,i}, z_{2,i-1}, \dots, z_{i,1}, z_{i+2,d-1}, \dots, z_{d,i+1}$ (note the missing $z_{i+1,d}$) to be mutually independent for each $i \in [d]$, then every entry in P can be viewed as chosen independently and uniformly at random. This can be verified in a straightforward way, and we can conclude the proof. \square

Remark 6.13. Following the similar argument, if we would like to get an $d \times d$ random matrix over \mathbb{F}_q , we can in turn do the following: take two $d \times d$ random alternating matrices X and Y and construct M as in Lemma 6.3. We then take another two random alternating matrices Z and W . We add up the first column of Z and W , of which each coordinates can be viewed as chosen independently and uniformly at random. We replace the last column of M by the new random vector, which gives an $d \times d$ random matrix.

We are now ready to prove Proposition 6.9.

Proof of Proposition 6.9. Given Proposition 6.11, we need upper bound the probability of a random $\mathbf{A} \in \Lambda(n, q)^c$, such that \mathbf{A} is *not* stable, by $1/q^{\Omega(n)}$.

By the union bound, we know that

$$\Pr[\mathbf{A} \in \Lambda(n, q)^c \text{ is not stable}] \leq \sum_{\substack{U \leq \mathbb{F}_q^n \\ 1 \leq \dim(U) \leq n-1}} \Pr[\mathbf{A} \in \Lambda(n, q)^c, \dim(\mathbf{A}(U)) \leq \dim(U)]. \quad (6.14)$$

We first simplify the right-hand-side. For a non-zero, proper $U \leq \mathbb{F}_q^n$, let $A_U := \{\mathbf{A} \in \Lambda(n, q)^r : \dim(\mathbf{A}(U)) \leq \dim(U)\}$. Clearly,

$$\Pr[\mathbf{A} \in \Lambda(n, q)^c, \dim(\mathbf{A}(U)) \leq \dim(U)] = \frac{|A_U|}{|\Lambda(n, q)^c|}.$$

We show that for any two dimension- d subspaces U and V , $|A_U| = |A_V|$. To see this, let $T \in \text{GL}(n, q)$ be any invertible matrix that sends V to U . Note that T further induces a linear map from $\Lambda(n, q)^r$ to itself by sending \mathbf{A} to $T^t \mathbf{A} T$. Since T is invertible, this map is a bijection. Moreover, for any $\mathbf{A} \in A_U$, we claim that $T^t \mathbf{A} T \in A_V$. This is because

$$\dim((T^t \mathbf{A} T)(V)) = \dim((T^t \mathbf{A})(U)) = \dim(\mathbf{A}(U)) \leq \dim(U) = \dim(V),$$

where the second equality holds since left and right multiplying invertible matrices does not change the rank of a matrix. To summarize, if $\dim(U) = \dim(V)$, then

$$\Pr[\mathbf{A} \in \Lambda(n, q)^c, \dim(\mathbf{A}(U)) \leq \dim(U)] = \Pr[\mathbf{A} \in \Lambda(n, q)^c, \dim(\mathbf{A}(V)) \leq \dim(V)].$$

The right-hand-side of 6.14 can be then simplified as

$$\Pr[\mathbf{A} \in \Lambda(n, q)^c \text{ is not stable}] \leq \sum_{d=1}^{n-1} \binom{n}{d}_q \cdot \Pr[\mathbf{A} \in \Lambda(n, q)^c, \dim(\mathbf{A}(U_d)) \leq d]. \quad (6.15)$$

where U_d is the d -dimensional subspace of \mathbb{F}_q^n spanned by the first d standard basis e_1, \dots, e_d .

The next goal is to upper bound $\left[\begin{smallmatrix} n \\ d \end{smallmatrix} \right]_q \Pr[\mathbf{A} \in \Lambda(n, q)^c, \dim(\mathbf{A}(U_d)) \leq d]$ for $d = 1, \dots, n-1$.

Let A_i^d be the $n \times d$ matrices consists of the first d columns of A_i for $i \in [c]$. (Note that the superscript here does not denote exponentiation.) Let $A^d = [A_1^d, \dots, A_c^d] \in M(n \times cd, q)$. Then $\dim(\mathbf{A}(U_d))$ is just the rank of A^d . Note that for $i \in [c]$, the first d row of A_i^d can be viewed as a random alternating matrix from $\Lambda(d, q)$, and the last $n-d$ rows of A_i^d can be viewed as a $(n-d) \times d$ random matrix. Moreover, these two matrices can be viewed as being chosen independently.

By Lemma 6.12 together with Remark 6.13, there exist a series of column operations represented by an invertible matrix $R \in GL(cd \times cd, q)$, such that the following holds. Let $V^d \in M(n \times 5d, q)$ be the matrix consists of the first $5d$ columns of $A^d R$. Then V^d can be viewed as chosen independently and uniformly at random from $M(n \times 5d, q)$, as \mathbf{A} is chosen uniformly at random from $\Lambda(n, q)^c$. Note that when $d = 1$, the first row of A_i^d is 0 for all $i \in [c]$. This degenerate case suggest us to consider V^1 as randomly choosing from $M((n-1) \times 5, q)$. Note that

$$\Pr[\mathbf{A} \in \Lambda(n, q)^c, \dim(\mathbf{A}(U_1)) \leq 1] \leq \Pr[V^1 \in M((n-1) \times 5, q), rk(V^1) \leq 1]$$

and

$$\Pr[\mathbf{A} \in \Lambda(n, q)^c, \dim(\mathbf{A}(U_d)) \leq d] \leq \Pr[V^d \in M(n \times 5d, q), rk(V^d) \leq d]$$

for $2 \leq d \leq n-1$.

We consider how to construct an $(n-1) \times 5$ matrix such that its rank is not larger than 1. One way to do so is to pick one column fix its coordinates; then let the rest 4 columns be scalar of the picked ones. This procedure gives the bound

$$\Pr[V^1 \in M((n-1) \times 5, q), rk(V^1) \leq 1] \leq \frac{\binom{5}{1} \cdot q^{n-1} \cdot q^{5-1}}{q^{5(n-1)}} = \frac{5}{q^{4n-8}}.$$

So we have

$$\left[\begin{smallmatrix} n \\ 1 \end{smallmatrix} \right]_q \cdot \Pr[\mathbf{A} \in \Lambda(n, q)^c, \dim(\mathbf{A}(U_1)) \leq 1] \leq \frac{5}{q^{3n-8}}. \quad (6.16)$$

Using the same idea, we deal with $2 \leq d \leq n-1$. All possible V^d such that $rk(V^d) \leq d$ can be constructed by first choosing d columns in V^d and fixing their entries, and then choosing the other columns from their linear span. This gives the bound

$$\Pr[V^d \in M(n \times 5d, q), rk(V^d) \leq d] \leq \frac{\binom{5d}{d} \times q^{nd} \times q^{4d^2}}{q^{5nd}} \leq \frac{1}{q^{4nd-4d^2-5d}},$$

where the last inequality uses $\binom{5d}{d} \leq 2^{5d} \leq q^{5d}$. For $d \leq \frac{n}{2}$, we upper bound $\left[\begin{smallmatrix} n \\ d \end{smallmatrix} \right]_q$ by q^{nd} . This gives that

$$\left[\begin{smallmatrix} n \\ d \end{smallmatrix} \right]_q \Pr[\mathbf{A} \in \Lambda(n, q)^c, \dim(\mathbf{A}(U_d)) \leq d] \leq \frac{1}{q^{3nd-4d^2-5d}} \leq \frac{1}{q^{6n-26}}. \quad (6.17)$$

For $\frac{n}{2} < d \leq n-2$, we upper bound $\left[\begin{smallmatrix} n \\ d \end{smallmatrix} \right]_q$ by $q^{n(n-d)}$. This gives that

$$\left[\begin{smallmatrix} n \\ d \end{smallmatrix} \right]_q \Pr[\mathbf{A} \in \Lambda(n, q)^c, \dim(\mathbf{A}(U_d)) \leq d] \leq \frac{1}{q^{5nd-n^2-4d^2-5d}} \leq \frac{1}{q^{n-6}}. \quad (6.18)$$

For $d = n-1$, we note that $\Pr[V^d \in M(n \times 5(n-1), q), rk(V^d) \leq n-1]$ is the probability that V^d is not of rank n when $n \geq 2$ [LQ, Fact 4 in arXiv version]. This gives the bound

$$\left[\begin{smallmatrix} n \\ n-1 \end{smallmatrix} \right]_q \Pr[\mathbf{A} \in \Lambda(n, q)^c, \dim(\mathbf{A}(U_d)) \leq d] \leq \frac{n \times n}{q^{5(n-1)-n+1}} = \frac{n^2}{q^{4(n-1)}}. \quad (6.19)$$

Combining equations from 6.14 to 6.19, we have

$$\begin{aligned} \Pr[\mathbf{A} \in \Lambda(n, q)^c \text{ is not stable}] &\leq \sum_{\substack{U \leq \mathbb{F}_q^n, \\ 1 \leq \dim(U) \leq n-1}} \Pr[\mathbf{A} \in \Lambda(n, q)^c, \dim(\mathbf{A}(U)) \leq d] \\ &\leq \sum_{d=1}^{n-1} \begin{bmatrix} n \\ d \end{bmatrix}_q \Pr[\mathbf{A} \in \Lambda(n, q)^c, \dim(\mathbf{A}(U_d)) \leq d] \leq \frac{1}{q^{\Omega(n)}}, \end{aligned}$$

which concludes the proof. \square

Remark 6.20 (Upgrading to the linear algebraic Erdős-Rényi model). In [LQ], the linear algebraic Erdős-Rényi model, $\text{LinER}(n, m, q)$, was introduced as the uniform distribution over all m -dimensional subspaces of $\Lambda(n, q)$. Randomly sampling m -tuples of $n \times n$ alternating matrices was termed as the naive model in [LQ]. It was also shown in [LQ] that the analysis in the naive model can be upgraded, with a mild loss in the parameters, to an analysis in $\text{LinER}(n, m, q)$. Such an upgrade can also be done similarly for the analysis here, though with a little bit more work than in [LQ]. We omit the details.

7 On testing isomorphism of groups with genus 2 radicals

In this section we show how to combine the methods of [GQ] for groups with abelian radicals and the methods of [BMW1] to study subclasses of groups whose solvable radicals are p -groups of class 2. Recall that p -groups of class 2 are considered as difficult as the general case for group isomorphism, so we did not expect to beat the $n^{\log n}$ bound for this entire class. However, as a corollary of the results in this section, we give an $n^{O(\log \log n)}$ -time isomorphism test for a class of groups whose radicals have genus 2. We shall work throughout with the following class of groups:

Let \mathcal{G} be the class of groups G whose solvable radical, $\text{Rad}(G)$, is a p -group of exponent $p \neq 2$ and class 2 upon which G acts as inner automorphisms of $\text{Rad}(G)$.

In [GQ] the classical strategy of using actions and cohomology was formally analyzed, showing that Gpl “splits” into two problems: Action Compatibility (ActComp), and Cohomology Class Isomorphism (Coholso); we state their definitions in the relevant sections below. When G has a normal subgroup N we may consider G as an extension of N by $Q = G/N$; both ActComp and Coholso have as their witnesses certainly elements of $\text{Aut}(N) \times \text{Aut}(Q) \times (Q \rightarrow N)$, and two groups are isomorphic if, and only if, there is a witness that works simultaneously for ActComp and Coholso (see [GQ] for a leisurely exposition). Furthermore, ActComp and Coholso each reduce to Gpl .

The two key cases to handle first are the extreme situations with regards to this natural splitting: semi-direct products, where the isomorphism problems reduce to just ActComp ; and “central” products (or rather, where $G/\text{Rad}(G)$ acts trivially on the radical $\text{Rad}(G)$), where the problem reduces to (nonabelian) Coholso . The class \mathcal{G} that we consider here is of the second type of extreme situation. We expect the first yield to techniques in [GQ, Section 3], perhaps using methods to solve isometry [IQ], but we are not yet able to see a clear path to this case.

7.1 Preliminaries on genus 2 groups

We briefly recall definitions and results on the automorphism group of groups of genus 2; see [BMW1] for details. For any group G , let $Z = Z(G)$ and $G' = [G, G]$; then we define the commutator map of G as $\circ_G: G/Z \times G/Z \rightarrow G'$. Two groups G, H are *isoclinic* if there are isomorphisms

$\varphi: G/Z(G) \rightarrow H/Z(H)$ and $\hat{\varphi}: G' \rightarrow H'$ such that $g_1^\varphi \circ_H g_2^\varphi = (g_1 \circ_G g_2)^{\hat{\varphi}}$. When G, H are nilpotent of class 2, their commutator maps are in fact \mathbb{Z} -bilinear (note that in this case $G/Z(G)$ is abelian), and the groups are isoclinic iff \circ_G and \circ_H are pseudo-isometric, by definition (recall §2). Given a bilinear map $\circ: U \times V \rightarrow W$ (U, V, W abelian groups), its *centroid* is

$$C(\circ) := \{(\varphi, \psi, \rho) \in \text{End}(U) \times \text{End}(V) \times \text{End}(W) : (\forall u \in U, v \in V)[u^\varphi \circ v = (u \circ v)^\rho = u \circ (v^\psi)]\};$$

the centroid is the largest ring of scalars over which \circ is bilinear. A nilpotent group G of class 2 is isoclinic to a direct product $H_1 \times \cdots \times H_s$ of directly indecomposable groups; the *genus* of G is the maximum rank of $[H_i, H_i]$ as a $C(\circ_{H_i})$ -module. Although the concept of genus is fully general, we focus on p -groups of exponent p and class 2; in this case isoclinism and isomorphism coincide, and centrally indecomposable p -groups of class 2 and exponent p have their centroids a finite field of characteristic p . For a biadditive map $\circ: U \times U \rightarrow V$, let $\Psi\text{Isom}(\circ)$ denote its group of pseudo-isometries; if \circ is bilinear over a field \mathbb{F} , let $\Psi\text{Isom}_{\mathbb{F}}(\circ) = \Psi\text{Isom}(\circ) \cap (\text{GL}_{\mathbb{F}}(U) \times \text{GL}_{\mathbb{F}}(V))$. Given a finite field \mathbb{F} of characteristic p , its *Galois group* denoted $\text{Gal}(\mathbb{F})$, consists of those field automorphisms of \mathbb{F} that act trivially on the prime subfield $\mathbb{Z}_p \leq \mathbb{F}$; $\text{Gal}(\mathbb{F})$ is cyclic of order $[\mathbb{F} : \mathbb{Z}_p] = \log_p |\mathbb{F}|$, generated by the Frobenius automorphism $a \mapsto a^p$.

Proposition 7.1 (See, e.g., [BMW1, Prop. 2.4]). *Let P be a p -group of class 2 and exponent p satisfying $Z(P) = [P, P]$. Then $\text{Aut}(P) = \Psi\text{Isom}(\circ_P) \rtimes \text{Hom}(P/Z(P), Z(P))$. If \circ_P is \mathbb{F} -bilinear, then $\Psi\text{Isom}_{\mathbb{F}}(\circ_P) \trianglelefteq \Psi\text{Isom}(\circ_P)$, with quotient $\Psi\text{Isom}(\circ_P)/\Psi\text{Isom}_{\mathbb{F}}(\circ_P) \cong \text{Gal}(\mathbb{F})$.*

Note that elements of $\Psi\text{Isom}_{\mathbb{F}}(\circ_P) \rtimes \text{Hom}(P/Z(P), Z(P))$ are faithfully represented by matrices $\begin{pmatrix} \alpha_V & d\alpha \\ 0 & \alpha_Z \end{pmatrix}$, where $\alpha_V \in \text{Aut}(P/Z(P))$, $\alpha_Z \in \text{Aut}(Z(P))$, and $d\alpha: P/Z(P) \rightarrow Z(P)$ is linear.

Recall that a map $\alpha: V \rightarrow W$ of \mathbb{F} -vector spaces is \mathbb{F} -semilinear if it is additive ($\alpha(v + v') = \alpha(v) + \alpha(v')$) and it is “twisted” linear, that is, $\alpha(\lambda v) = \lambda^\gamma \alpha(v)$, where $\gamma \in \text{Gal}(\mathbb{F})$. From the preceding, it follows immediately that:

Observation 7.2. *Let P be a p -group of class 2 and exponent p such that \circ_P is \mathbb{F} -bilinear. For any $\alpha \in \text{Aut}(P)$, the induced automorphisms on $[P, P]$ and $P/[P, P]$ are both \mathbb{F} -semilinear.*

Observation 7.3. *If P is a p -group of class 2 and exponent p such that $Z(P) \neq [P, P]$, then $P \cong Q \times A$, where Q is characteristic subgroup of P and satisfies $Z(Q) = [Q, Q]$, and A is an elementary abelian p -group. Moreover, Q and A and the isomorphism $P \cong Q \times A$ can be constructed in polynomial time in the number of generators, even when the groups are given as a black box.*

Standard proof sketch. $Z(P) \geq [P, P]$ since P is of class 2. Since P is of exponent p , $Z(P)$ is elementary abelian, and thus is a vector space \mathbb{Z}_p^e . Let $\{g_1, \dots, g_s\}$ be a generating set of P . Let $Q = \langle g_i : g_i \notin Z(P) \rangle$. Then $Q \cap Z(P) = [P, P]$. Let A be a \mathbb{Z}_p -linear complement to $[P, P]$ in $Z(P)$. \square

Theorem 7.4 ([BMW1, IQ]). *Let P be a p -group of class 2, exponent $p \neq 2$, and genus g . Given $\alpha \in \text{Aut}(Z(P))$, one can test whether α extends to an automorphism $\hat{\alpha} \in \text{Aut}(P)$ in poly-logarithmic time when $g \leq 2$, and in polynomial time otherwise.*

Proof. When $g = 2$, the result is immediate from [BMW1, Thm. 3.22], and their comments about its constructive nature (see [BMW1, §6.2]). In general, this is an isometry problem, which is solvable in polynomial time [IQ]. \square

Theorem 7.5. *Isomorphism of p -groups of class 2, exponent $p \neq 2$, and genus ≤ 2 can be decided in poly-logarithmic time [BMW1, Thm. 1.1], and of genus $\leq \sqrt{\log |G|}$ can be decided in polynomial time [IQ, Thm. 3].*

7.2 Testing isomorphism in the class \mathcal{G}

Our goal in this final section is to prove Theorem F, which for convenience we now recall:

Theorem F. *Let \mathcal{G} be the class of groups G defined at the start of Section 7. Given groups G_1, G_2 of order n , it can be decided in $\text{poly}(n)$ time if they lie in \mathcal{G} . If so, isomorphism can be decided, and a generating set for $\text{Aut}(G_i)$ found, in time $n^{O(g+\log \log n)}$, where g is the genus of $\text{Rad}(G)$.*

We will need the following two results from Grochow–Qiao [GQ], which first require a few concepts we haven't yet discussed. Recall that a pair of subgroups $H_1, H_2 \leq G$ is a *central decomposition* of G if $\langle H_1, H_2 \rangle = G$ and $[H_1, H_2] = 1$. Given two groups M_1, M_2 and an isomorphism $\varphi: Y_1 \rightarrow Y_2$ between two subgroups $Y_i \leq Z(M_i)$, the quotient of $M_1 \times M_2$ by $\{(y^{-1}, \varphi(y)) : y \in Y_1\}$ is the *central product* of M_1 and M_2 along φ , denoted $M_1 \times_{\varphi} M_2$, and φ is called the *amalgamating map*. In this case, $\{M_1, M_2\}$ is a central decomposition of $M_1 \times_{\varphi} M_2$; conversely, if $\{H_1, H_2\}$ is a central decomposition of a group G , then there exist $Y_i \leq Z(H_i)$ and an isomorphism $\varphi: Y_1 \rightarrow Y_2$ such that $G \cong H_1 \times_{\varphi} H_2$.

Lemma 7.6 ([GQ, Lem. 3.10]). *Let $N \trianglelefteq G$, and suppose G acts on N as inner automorphisms of N . Then there is a subgroup $H \leq G$, constructible in time $\text{poly}(|G|)$, such that $H \cap N = Z(N)$, $H/N = Q$, and $\{N, H\}$ is a central decomposition of G . We denote this subgroup H by $G|_{Z(N)}$.*

Proposition 7.7 (Special case of [GQ, Prop. 3.13]). *Let G_i ($i = 1, 2$) be a group such that $\text{Rad}(G_i) = P$ is a p -group of class 2, exponent p , and genus 2, and such that $Q = G_i / \text{Rad}(G_i)$ acts on $\text{Rad}(G_i)$ by inner automorphisms of $\text{Rad}(G_i)$. Suppose that $G_1|_{Z(P)} \cong G_2|_{Z(P)}$ (as in Lem. 7.6), which we denote by \hat{Q} , and let $\varphi_i: Z(P) \rightarrow Z(\hat{Q})$ be the corresponding amalgamating maps. Then $G_1 \cong G_2$ iff there exist $(\alpha, \beta) \in \text{Aut}(P) \times \text{Aut}(\hat{Q})$ such that $\varphi_1 = \beta^{-1}|_{Z(\hat{Q})} \circ \varphi_2 \circ \alpha|_{Z(P)}$.*

Proposition 7.8 (See [GQ, §6.1.2, p. 1186]). *Let G be a group with $\text{Rad}(G) = Z(G)$, and let $Q = G/Z(G)$ an elementary abelian group. Given $\beta \in \text{Aut}(Q)$, one can compute in $\text{poly dim } Z(G)$ time a single $\alpha \in \text{Aut}(Z(G))$ and a basis of a linear subspace $L \subseteq \text{End}(Z(G))$ such that $(\beta, \gamma) \in \text{Aut}(G)$ iff $\gamma \in \alpha + L$.*

Proof of Thm. F. Let G_1, G_2 be groups satisfying the hypotheses. In $\text{poly}(|G|)$ time, find $\text{Rad}(G_i)$ and denote this by P'_i . By Lem. 7.6, construct $\hat{Q}_i = G_i|_{Z(P'_i)}$ and the amalgamating maps $\varphi'_i: Z(P'_i) \rightarrow Z(\hat{Q}_i)$. Using Thm. 7.5 [BMW1, IQ], decide whether $P'_1 \cong P'_2$; if not, then $G_1 \not\cong G_2$ and we can stop, and if so, then let $\rho': P'_1 \rightarrow P'_2$ be such an isomorphism.

Note (Observation 7.3) that it may be the case that $P'_i \cong P_i \times A_i$ for some abelian groups A_i ; if this is the case, we can find P_i and A_i such that $Z(P_i) = [P_i, P_i]$ in polynomial time. Replace P'_i by P_i and replace ρ' by $\rho := \rho'|_{P_1}$; this will not hurt us later because P_i is characteristic in P'_i , and therefore also in G_i . Intuitively, the only place that A_i interacts with P'_i is as a direct product, and the only way A_i interacts with \hat{Q}_i is as a subgroup of its center, where A_i still appears.

Next, since \hat{Q}_i is a group with $\text{Rad}(\hat{Q}_i) \leq Z(\hat{Q}_i)$, by [GQ] we can decide whether $\hat{Q}_1 \cong \hat{Q}_2$ in time $n^{O(\log \log n)}$; if not, then $G_1 \not\cong G_2$ and we can stop, and if so, let $\tau: \hat{Q}_1 \rightarrow \hat{Q}_2$ be such an isomorphism. Let $\varphi_1 = \varphi'_1$ and $\varphi_2 = \tau^{-1} \circ \varphi'_2 \circ \rho^{-1}$. These are both isomorphisms $Z(P_1) \rightarrow Z(\hat{Q}_1)$, so from now on we let $P = P_1$ and $\hat{Q} = \hat{Q}_1$, and we have $G_i \cong P \times_{\varphi'_i} \hat{Q}$ for $i = 1, 2$.

Now, by Proposition 7.7, $G_1 \cong G_2$ iff there exists $(\alpha, \beta) \in \text{Aut}(P) \times \text{Aut}(\hat{Q})$ such that

$$\varphi'_1 = \beta^{-1}|_{Z(\hat{Q})} \circ \varphi'_2 \circ \alpha|_{Z(P)}. \quad (7.9)$$

By Observation 7.2, $\alpha|_{Z(P)}$ is \mathbb{F} -semilinear, and since P has genus g , $Z(P) \cong \mathbb{F}^g$. Enumerate $\text{GL}(\mathbb{F}^g)$; for each $\alpha \in \text{GL}(\mathbb{F}^g)$, check whether α extends to an automorphism of P (Theorem 7.4

[BMW1, IQ]). Let $Q = \hat{Q}/Z(\hat{Q}) = \hat{Q}/\text{Rad}(\hat{Q})$. Enumerate $\gamma \in \text{Aut}(Q)$. For each $\alpha \in \Gamma\text{L}(\mathbb{F}^g)$ that extends to an automorphism of P , and each $\gamma \in \text{Aut}(Q)$, we seek $\beta \in \text{Aut}(Z(\hat{Q}))$ such that (γ, β) induces an automorphism of \hat{Q} and (α, β) satisfies (7.9). By Proposition 7.8, the set of such γ such that (γ, β) is an automorphism of \hat{Q} is an affine linear space $\beta_0 + B$, where B is a linear subspace of $\text{End}(Z(\hat{Q}))$, and we can compute γ_0 and a basis for B in polynomial time. Once α is fixed, (7.9) is linear in β . Intersecting the linear space which solves (7.9) with the affine space $\beta_0 + B$ is standard linear algebra, and can thus be computed in polynomial time.

To summarize, for each $\alpha \in \text{Aut}_{\mathbb{F}}(Z(P)) \cong \Gamma\text{L}_g(\mathbb{F})$ and each $\gamma \in \text{Aut}(Q)$, we can compute a single element and generating set for those β such that α extends to an automorphism P , $(\beta, \gamma) \in \text{Aut}(\hat{Q})$, and (α, β) satisfy (7.9). Taking the union over all choices in $\Gamma\text{L}_g(\mathbb{F})$ and $\text{Aut}(Q)$ gives us the coset of isomorphisms $G_1 \rightarrow G_2$.

Analysis of running time. When $g \leq O(\log \log |G|)$, we have $|\Gamma\text{L}_g(\mathbb{F})| \sim |\text{Gal}(\mathbb{F})| \cdot \mathbb{F}^{g^2} = k(p^k)^{g^2} = k(p^{kg})^g = k|Z(P)|^g \leq |G|^{g+o(1)}$ where $|G| \geq |\mathbb{F}| = p^k$, so their number is not too large, and $\Gamma\text{L}_g(\mathbb{F})$ is easily enumerated in $n^{O(g)}$ time. By [BCGQ], $\text{Aut}(Q)$ can be listed in time $n^{O(\log \log n)}$. Since we are enumerating over both of these, we take their product $n^{O(g+\log \log n)}$, which ends up dominating the runtime. By [GQ], isomorphism of \hat{Q}_1 and \hat{Q}_2 can be tested in $n^{O(\log \log n)}$ time. The rest is polynomial time or poly-logarithmic time by previous results, or linear algebra (poly-logarithmic time in $|G|$). \square

Remark 7.10. There is some hope when $g \leq 2$ in Theorem F—due to the poly-logarithmic isomorphism test of [BMW1]—to improve this poly-logarithmic time. However, a prerequisite is first solving isomorphism of groups with no abelian normal subgroups in poly-logarithmic time, rather than just polynomial [BCQ].

Acknowledgments

The authors would like to acknowledge V. Arvind and M. Grohe for useful comments on hypergraph k -WL, Avinoam Mann for discussions on random generation of p -groups, and László Babai and Xiaorui Sun for discussions on average-case algorithms for testing isomorphism of p -groups of class 2 and exponent p . P. A. B. was partially supported by NSF grant DMS-1620362. J. A. G. was partially supported by NSF grant DMS-1750319. Y. L. was partially supported by ERC Consolidator Grant 615307-QPROGRESS. Y. Q. was partially supported by the Australian Research Council DECRA DE150100720. J. B. W. was partially supported by NSF grant DMS-1620454. P. A. B. and J. B. W. also acknowledge the Hausdorff Institute for Mathematics, and the University of Auckland where some of this research was conducted. P. A. B., J. A. G., J. B. W., and Y. Q. also acknowledge the Santa Fe Institute, where some of this research was conducted.

References

- [A] Sergei I. Adian, *Unsolvability of some algorithmic problems in the theory of groups*, Trudy Moskovskogo Matematicheskogo Obshchestva **6** (1957), 231–298.
- [AFKV] V. Arvind, Frank Fuhlbrück, Johannes Köbler, and Oleg Verbitsky, *On Weisfeiler–Leman invariance: Sub-graph counts and related graph properties*. arXiv:1811.04801, 2018.
- [B1] László Babai, *Graph isomorphism in quasipolynomial time [extended abstract]*, Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, pp. 684–697. arXiv:1512.03547, version 2.
- [B2] ———, *Lecture on graph isomorphism*, 1979.

- [B3] Reinhold Baer, *Groups with abelian central quotient group*, Transactions of the American Mathematical Society **44** (1938), no. 3, 357–386.
- [BB] László Babai and Robert Beals, *A polynomial-time theory of black box groups I*, London Mathematical Society Lecture Note Series (1999), 30–64.
- [B1] Jan Böker, *Color refinement, homomorphisms, and hypergraphs*. arXiv: 1903.12432, 2019.
- [B2] William W. Boone, *The word problem*, Annals of Mathematics (1959), 207–265.
- [B3] H.R. Brahana, *Metabelian groups and trilinear forms*, Duke Mathematical Journal **1** (1935), no. 2, 185–197.
- [BCGQ] László Babai, Paolo Codenotti, Joshua A. Grochow, and Youming Qiao, *Code equivalence and group isomorphism*, Proceedings of the Twenty-Second Annual ACM–SIAM Symposium on Discrete Algorithms SODA 2011, pp. 1395–1408.
- [BCQ] László Babai, Paolo Codenotti, and Youming Qiao, *Polynomial-time isomorphism test for groups with no abelian normal subgroups - (extended abstract)*, Automata, languages, and programming - 39th international colloquium, ICALP 2012, pp. 51–62.
- [BES] László Babai, Paul Erdős, and Stanley M. Selkow, *Random graph isomorphism*, SIAM J. Comput. **9** (1980), no. 3, 628–635.
- [BJP] W. Bosma, J. J. Cannon, and C. Playoust, *The Magma algebra system I: the user language*, J. Symb. Comput. (1997), 235–265.
- [BMW1] Peter A. Brooksbank, Joshua Maglione, and James B. Wilson, *A fast isomorphism test for groups whose Lie algebra has genus 2*, J. Algebra **473** (2017), 545–590.
- [BMW2] ———, *Thetensor.space*, GitHub, 2019.
- [BNV] Simon R. Blackburn, Peter M. Neumann, and Geetha Venkataraman, *Enumeration of finite groups*, Cambridge Tracts in Mathematics, vol. 173, Cambridge University Press, Cambridge, 2007.
- [BOW] Peter A. Brooksbank, Eamonn A. O’Brien, and James B. Wilson, *Isomorphism testing of graded algebras*. arXiv:1708.08873, 2017.
- [BS] L. Babai and E. Szemerédi, *On the complexity of matrix group problems I*, Proceedings of the 25th annual symposium on foundations of computer science, SFCS 1984, pp. 229–240.
- [BW] Peter A. Brooksbank and James B. Wilson, *Computing isometry groups of Hermitian maps*, Trans. Amer. Math. Soc. **364** (2012), no. 4, 1975–1996.
- [CFI] Jin-Yi Cai, Martin Fürer, and Neil Immerman, *An optimal lower bound on the number of variables for graph identifications*, Combinatorica **12** (1992), no. 4, 389–410.
- [CH] John Cannon and Derek F. Holt, *Automorphism group computation and isomorphism testing in finite groups*, J. Symbolic Comput. **35** (2003), no. 3, 241–267.
- [DGR] Holger Dell, Martin Grohe, and Gaurav Rattan, *Lovász meets Weisfeiler and Leman*, 45th international colloquium on automata, languages, and programming, ICALP 2018, pp. 40:1–40:14.
- [FSEDS] Jean-Charles Faugère, Mohab Safey El Din, and Pierre-Jean Spaenlehauer, *On the complexity of the generalized MinRank problem*, J. Symbolic Comput. **55** (2013), 30–58.
- [G1] Robert Gilmon, *Algorithmic search in group theory*. arXiv:1812.08116, 2018.
- [G2] M. Gromov, *Random walk in random groups*, Geom. Funct. Anal. **13** (2003), no. 1, 73–146.
- [GQ] Joshua A. Grochow and Youming Qiao, *Algorithms for group isomorphism via group extensions and cohomology*, SIAM J. Comput. **46** (2017), no. 4, 1153–1216.
- [H1] J. Harris, *Algebraic geometry*, Graduate Texts in Mathematics 133 (1992).
- [H2] Graham Higman, *Enumerating p -groups. I: Inequalities*, Proceedings of the London Mathematical Society **3** (1960), no. 1, 24–30.
- [HL] Hermann Heineken and Hans Liebeck, *The occurrence of finite groups in the automorphism group of nilpotent groups of class 2*, Arch. Math. (Basel) **25** (1974), 8–16.
- [HM] Geir T. Helleloid and Ursula Martin, *The automorphism group of a finite p -group is almost always a p -group*, J. Algebra **312** (2007), no. 1, 294–329.
- [HR] Derek F. Holt and Sarah Rees, *Testing modules for irreducibility*, J. Austral. Math. Soc. **1** (1994), 1–16.
- [IL] Neil Immerman and Eric S. Lander, *Describing graphs: a first-order approach to graph canonization*, Complexity theory retrospective—in honor of Juris Hartmanis on the occasion of his 60th birthday, 1990, pp. 59–81.

- [IQ] 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*, Proceedings of the twenty-ninth annual ACM-SIAM symposium on discrete algorithms, SODA 2018, pp. 2357–2376.
- [KL] William M. Kantor and Alexander Lubotzky, *The probability of generating a finite classical group*, *Geom. Dedicata* **36** (1990), no. 1, 67–87.
- [KS] Neeraj Kayal and Nitin Saxena, *Complexity of ring morphism problems*, *Computational Complexity* **15** (2006), no. 4, 342–390.
- [KTW] Martin Kassabov, Brady Tyburski, and James B. Wilson, *The number of isomorphism types of subgroups of simple groups is maximum possible*. (in preparation).
- [LGR] Francois Le Gall and David J. Rosenbaum, *On the group and color isomorphism problems*. arXiv:1609.08253, 2016.
- [L1] Eugene M. Luks, *Group isomorphism with fixed subnormal chains*. arXiv: 1511.00151, 2015.
- [L2] ———, *Hypergraph isomorphism and structural equivalence of boolean functions*, Proceedings of the thirty-first annual ACM symposium on theory of computing STOC 1999, pp. 652–658.
- [L3] ———, *Permutation groups and polynomial-time computation*, Groups and Computation, 1993.
- [LQ] Yinan Li and Youming Qiao, *Linear algebraic analogues of the graph isomorphism problem and the erdős-rényi model*, 58th IEEE annual symposium on foundations of computer science, FOCS 2017, pp. 463–474.
- [LV] Ruvim Lipyanski and Natalia Vanetik, *On Borel complexity of the isomorphism problems for graph related classes of Lie algebras and finite p -groups*, *J. Algebra Appl.* **14** (2015), no. 5, 1550078, 15.
- [LW1] Serge Lang and André Weil, *Number of points of varieties in finite fields*, *American Journal of Mathematics* **76** (1954), no. 4, 819–827.
- [LW2] Mark L. Lewis and James B. Wilson, *Isomorphism in expanding families of indistinguishable groups*, *Groups Complex. Cryptol.* **4** (2012), no. 1, 73–110.
- [M1] Joshua Maglione, *Compatible filters for isomorphism testing*. arXiv:1805.03732, 2018.
- [M2] ———, *Longer nilpotent series for classical unipotent subgroups*, *J. Group Theory* **18** (2015), no. 4, 569–585. MR3365818
- [M3] ———, *Efficient characteristic refinements for finite groups*, *J. Symbolic Comput.* **80** (2017), 511–520.
- [M4] Avinoam Mann, *Some questions about p -groups*, *J. Austral. Math. Soc. Ser. A* **67** (1999), no. 3, 356–379.
- [N] P.S. Novikov, *On algorithmic undecidability of the word problem in the theory of groups*, *Trudy Mat. Inst. Steklov* **44** (1955), 1–144.
- [R1] Michael O. Rabin, *Recursive unsolvability of group theoretic problems*, *Annals of Mathematics* (1958), 172–194.
- [R2] David J. Rosenbaum, *Bidirectional collision detection and faster deterministic isomorphism testing*. arXiv: 1304.3935, 2013.
- [R3] ———, *Breaking the $n \log n$ barrier for solvable-group isomorphism*, Proceedings of the Twenty-fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 13, pp. 1054–1073.
- [RW] David J. Rosenbaum and Fabian Wagner, *Beating the generator-enumeration bound for p -group isomorphism*, *Theoret. Comput. Sci.* **593** (2015), 16–25.
- [S] Charles C. Sims, *Enumerating p -groups*, *Proceedings of the London Mathematical Society* **3** (1965), no. 1, 151–166.
- [V] A.M. Vershik, *Statistical mechanics of combinatorial partitions, and their limit shapes*, V.A. Steklov Institute of Mathematics, Russian Academy of Sciences.
- [WL] Boris Weisfeiler and Andrei A. Lehman, *A reduction of a graph to a canonical form and an algebra arising during this reduction*, *Nauchno-Technicheskaya Informatsia* **2** (1968), no. 9, 12–16. English translation available at <https://www.iti.zcu.cz/wl2018/wlpaper.html>.
- [W1] B. Weisfeiler (ed.), *On construction and identification of graphs*, *Lecture Notes in Mathematics*, vol. 558, Springer-Verlag, 1976. With contributions by A. Lehman, G. M. Adelson-Velsky, V. Arlazarov, I. Faragev, A. Uskov, I. Zuev, M. Rosenfeld, and B. Weisfeiler.
- [W2] James B. Wilson, *The threshold for subgroup profiles to agree is $\Omega(\log n)$* . arXiv:1612.01444.
- [W3] ———, *More characteristic subgroups, Lie rings, and isomorphism tests for p -groups*, *J. Group Theory* **16** (2013), no. 6, 875–897. MR3198722
- [W4] ———, *Skolem–Noether for nilpotent products*, arXiv preprint arXiv:1507.04406 (2015).