12-2019

Testing isomorphism of graded algebras

Peter A. Brooksbank
pbrooksb@bucknell.edu

James B. Wilson
Colorado State University - Fort Collins, james.wilson@colostate.edu

Eamonn A. O'Brien
University of Auckland, e.obrien@aucklan.ac.nz

TESTING ISOMORPHISM OF GRADED ALGEBRAS

PETER A. BROOKSBANK, E. A. O'BRIEN, AND JAMES B. WILSON

Abstract. We present a new algorithm to decide isomorphism between finite graded algebras. For a broad class of nilpotent Lie algebras, we demonstrate that it runs in time polynomial in the order of the input algebras. We introduce heuristics that often dramatically improve the performance of the algorithm and report on an implementation in Magma.

1. Introduction

It is possible to decide if algebraic objects A and B of order n are isomorphic by fixing a generating sequence $a_1, \ldots, a_d$ for A and searching through all sequences $b_1, \ldots, b_d$ in B until we find an identification $a_i \mapsto b_i$ that extends to an isomorphism $A \to B$. This process takes n^d steps: for groups and algebras, where d can be as large as $\log n$, the resulting complexity is not polynomial in the orders of the input objects. Despite significant progress over the years on various isomorphism problems, asymptotic improvements over "brute force" for substantial classes of objects are rare.

We introduce a general strategy for testing isomorphism of finite graded algebras. It is particularly effective for nilpotent matrix Lie algebras, and we describe a class of such algebras for which our isomorphism test runs in time polynomial in the order of the input algebra. We have also implemented a version in Magma [3].

While graded algebras are natural structures in their own right, they also arise from the study of other algebraic structures. For example, given a ring R, one can compute its Jacobson radical, J, and consider the graded algebra $\operatorname{gr} R = R/J \oplus \bigoplus_{i=1}^{\infty} J^i/J^{i+1}$. Similarly, the intersection $O_p(G)$ of the Sylow p-subgroups of a finite group G for a prime p dividing $|G|$ is normal and nilpotent, so $\operatorname{gr} G = \mathbb{Z}[G/O_p(G)] \oplus \bigoplus_{i=1}^{\infty} \eta_i/\eta_{i+1}$ is a graded Lie $\mathbb{F}_p$-algebra, where $\eta_1 := O_p(G)$ and $\eta_{i+1} = [\eta_i, \eta_1]\eta_i^p$. Isomorphism tests for associated graded structures work with individual graded components, often exploiting the power of linear algebra.

Existing uses of graded algebras within isomorphism testing proceed sequentially through the grading; see [12], for example. Starting with the first graded component, one considers all possible isomorphisms between corresponding components, and uses the graded product to decide which of them induces an isomorphism between subsequent components. While this iterative approach usually offers improvements over brute force, a single large homogeneous component may create a bottleneck. Our approach is not constrained by the need to process the components sequentially. Instead, it identifies sections of the two graded algebras between which

This work was supported in part by the Marsden Fund of New Zealand via grant UOA 1626, by NSF grants DMS-1620454 and DMS-1620362, and by the Simons Foundation #281435. We thank the referee for helpful comments.

the list of possible maps is small and can be computed quickly. It then determines which of these maps between sections lift to isomorphisms of the algebras.

To state our main result, we require a few preliminaries. Let K be a finite field. A *K-algebra* is a K-module, A, equipped with a (possibly nonassociative) K-bilinear product $\circ\colon A \times A \rightarrowtail A$. If, as a K-module,

$$A = \bigoplus_{s=0}^{\infty} A_s, \text{ where } A_s \circ A_t \leqslant A_{s+t},$$

then A is *$\mathbb{N}$-graded*. We assume that A is *generated in degree* 1 in the sense that, for all $s > 0$, $A_s = \sum_{j=1}^{s-1} A_j \circ A_{s-j}$. An isomorphism between graded algebras that maps each graded component of one algebra to the corresponding component of the other is a *graded isomorphism*. For each $s \geqslant 1$, one can restrict the product on A to produce a *bilinear map* $A_1 \times A_s \rightarrowtail A_{s+1}$. The ring of *adjoints* of this bilinear map, denoted $\mathcal{M}_s$, is the largest ring R faithfully represented on $A_1 \oplus A_s$ such that the bilinear map $A_1 \times A_s \rightarrowtail A_{s+1}$ factors through the tensor product space $A_1 \otimes_R A_s$. Defined explicitly as operators in (2.7), $\mathcal{M}_s$ has group of units $\mathcal{M}_s^{\times}$.

Our main result, proved in Section 4, is the following.

Theorem 1.1. *There is a deterministic algorithm that, given two finite graded K-algebras A and B generated in degree 1, decides whether or not there is a graded isomorphism $A \to B$. The algorithm has complexity*

$$O(\min_s\{|\mathcal{M}_s^{\times}| \cdot |\operatorname{Aut}(A_{s+1})|\} \cdot (\dim A)^{2\omega} \cdot \log^2 |K|),$$

where s runs over the grading and $2 \leqslant \omega < 3$ is the exponent of matrix multiplication over K.

By comparison, the sequential approach has complexity $\tilde{O}(|K|^{(\dim A)^2})$. While the estimate in Theorem 1.1 is not asymptotically better in all cases, the flexibility to choose which component to process first may lead to dramatic improvements. As one illustrative example, define $\mathcal{F}_q(d_1, \ldots, d_\ell)$ to be the smallest class of graded Lie algebras containing the Lie subalgebras $\mathfrak{L}_* \leqslant \mathfrak{gl}_{d_1+\ldots+d_\ell}(q)$ satisfying

$$[\mathfrak{L}_*, \mathfrak{L}_*] = \left\{ \begin{bmatrix} 0_{d_1} & 0_{d_1,d_2} & * & \cdots & * \\ & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & * \\ & & & 0_{d_{\ell-1}} & 0_{d_{\ell-1},d_\ell} \\ & & & & 0_{d_\ell} \end{bmatrix} \right\} \leqslant \mathfrak{L}_* \leqslant \left\{ \begin{bmatrix} 0_{d_1} & * & \cdots & * \\ & \ddots & \ddots & \vdots \\ & & \ddots & * \\ & & & 0_{d_\ell} \end{bmatrix} \right\},$$

where $0_{m,n}$ denotes the zero $(m \times n)$-matrix, and $0_n = 0_{n,n}$. (A *class of algebras* is a collection closed under isomorphism; in particular, no specific representation is assumed as part of the input.) An analysis of the algorithm in Theorem 1.1 applied to $\mathcal{F}(d_1, \ldots, d_\ell)$ establishes the following.

Theorem 1.2. *Isomorphism testing in $\mathcal{F}_q(d_1, \ldots, d_\ell)$ is in time $\tilde{O}(|\mathfrak{L}|^{m^2/\varepsilon})$, where*

$$\varepsilon = \sum_{1 \leqslant i < j \leqslant \ell} d_i d_j, \qquad m = \min_{s \leqslant 1+\frac{d}{2}} \left\{ \sum_{i=1}^{\ell-s} d_i d_{i+s} \right\}.$$

The number of isomorphism classes in $\mathcal{F}(d_1, \ldots, d_\ell)$ is $q^{O((d_1+\cdots+d_\ell)^2)}$.

The complexity in Theorem 1.2 is polynomial in the size of the algebras when both $m^2, \varepsilon \in O((d_1 + \ldots + d_\ell)^2)$, a condition that holds, for instance, when d_i is bounded, and in many other cases.

One of the motivations of this work is to develop practical tools to decide isomorphism within families of finite groups and algebras. From this viewpoint, we are concerned with developing algorithms that perform well as a function of the length of standard encodings of the input algebras, such as by generating sets, or as bases with structures constants. Despite the significant improvements offered by the strategy underlying our main results, we have encountered situations where the necessary exhaustive search is still intractably large.

To address this concern, in Section 7 we introduce heuristics that use local invariants to deduce global restrictions on the possible automorphisms arising from $A_1 \times A_s \rightarrowtail A_{s+1}$, thereby reducing substantially the ensuing exhaustive search. More precisely, we design a labeling of the points and lines of the projective space on A_{s+1} that is invariant under isomorphism. The labeling offers sufficient variability that the resulting constraints often reduce intractable searches to the practical realm. We also revisit another "local-to-global" process called *fingerprinting* that was introduced in [13]. In Section 8 we report on our implementation of the algorithms in Magma, and show for some families of examples that our techniques have significant practical impact.

We prove a more general version of Theorem 1.1 in which algebras are not necessarily generated in degree 1, and may be graded using an arbitrary monoid. This general treatment allows our algorithms to take as input more refined gradings on the given algebras that often decompose them into smaller pieces. The development of such refined gradings is an emerging area of study that may lead to faster isomorphism tests; see [16], for example.

2. Preliminaries

Throughout the paper K denotes a finite field, and all K-vector spaces are finite-dimensional. The K-dual of a K-vector space V is denoted by $V^\dagger$. If $f\colon U \to V$ is a linear map, then $f^\dagger\colon V^\dagger \to U^\dagger$ denotes the dual map.

2.1. **Graded algebras.** For convenience in our exposition we assume that an algebra A is specified by a basis $\{a_1, \ldots, a_d\}$ together with *structure constants* $[\alpha_{ij}^k]$ defined by

$$a_i \cdot a_j = \sum_k \alpha_{ij}^k a_k.$$

Although the structure constant model of an algebra is both less compact and less efficient than alternatives—such as matrix algebras specified by generating sets, multivariate polynomials, and more general quotients of free algebras—it provides a convenient, uniform starting point.

Let M be a commutative monoid with pre-order $\prec$, where $a \prec b$ if there exists $c \in M$ such that $a + c = b$. We assume that, relative to $\prec$, every nonempty subset has a minimal element. We also assume that 0 is a minimal element of M so that M is conical. This ensures that we can perform (Noetherian) induction on the indices in M: if $S \subset M$ has the property that for every $s \in S$ there exists $t \in S \setminus \{s\}$ with $t \prec s$, then $S = \emptyset$. The monoids $M = \mathbb{N}^c$ satisfy the necessary conditions.

An algebra A_* is *M-graded* if $A_* = \bigoplus_{s\in M} A_s$ and for all $s, t \in M$, $A_s \circ A_t \subset A_{s+t}$. We assume, for each s, that A_s is the K-linear span of $A_s \cap \{a_1, \ldots, a_d\}$. We say that A is *generated in degrees T* if

$$(\forall s) \qquad s \in M \setminus T \implies \sum_{\substack{s = s_1 + s_2,\\ s_1, s_2 \notin \{0, s\}}} A_{s_1} \circ A_{s_2} \;=\; A_s. \tag{2.1}$$

As M is conical, if A_* is generated in degrees T, then $0 \in T$. The following observation is a direct consequence of the induction principle.

Lemma 2.2. *If A_* is an M-graded algebra generated in degrees T then, for each $s \in M$, A_s consists of linear combinations of products of elements in $\bigcup_{t\in T} A_t$.*

If $M = \mathbb{N}$, then the generating degrees are 0 and 1. Here, A_0 contributes no shift in the grading; it is customary to ignore 0 and regard A_* as generated in degree 1. The main results in the introduction were formulated for this special case.

As an infinite monoid can grade a finite algebra, repetitions $A_s = A_t$ for $s \neq t$ are common. To avoid redundancy, one can truncate all rays in the conical monoid M in the first place where all are stable. This is always possible but may produce a less familiar monoid: for each $c \in \mathbb{N}$, associate to M the ℓ-truncated cyclic monoid $\mathbb{N}_\ell = \{0, \ldots, \ell\}$, where $x \boxplus y := x + y$ if $x + y < \ell$, and $x \boxplus y := \ell$ otherwise.

Proposition 2.3. *Let $M = \langle m_1, \ldots, m_d\rangle$, let A_* be an M-graded algebra, and let ℓ be the smallest positive integer satisfying*

$$\forall i \in \{1, \ldots, d\},\ \forall u \in M \setminus \{0\} \qquad A_{\ell m_i} = A_{\ell m_i + u}.$$

Then A_* is naturally $(\mathbb{N}_\ell)^d$–graded where $A_{(r_1,\ldots,r_d)} = A_{r_1 m_1 + \cdots + r_d m_d}$. In particular, we may assume for fixed d that $|M| \in O((\dim A_*)^d)$.

2.2. **Bimaps.** A *bi-additive map* (or just *bimap*) is a tuple $U_* = \langle U_2, U_1, U_0, \circ\rangle$ where the U_i are abelian groups and $\circ\colon U_2 \times U_1 \rightarrowtail U_0$ is a function satisfying the following two-sided distributive law:

$$(u_2 + u_2') \circ u_1 = u_2 \circ u_1 + u_2' \circ u_1 \qquad u_2 \circ (u_1 + u_1') = u_2 \circ u_1 + u_2 \circ u_1'.$$

We always work with K-bimaps: the U_i are K-modules and $\circ$ is K-bilinear. The *dimension* of U_* is the sum of the dimensions of its component spaces, namely

$$\dim U_* := \dim U_2 + \dim U_1 + \dim U_0.$$

A *homotopism* $f_*\colon U_* \to V_*$ is a tuple $(f_i\colon U_i \to V_i \mid i \in \{0, 1, 2\})$ satisfying

$$(\forall u_2)(\forall u_1) \qquad (u_2 \circ u_1)f_0 = u_2 f_2 \circ u_1 f_1. \tag{2.4}$$

Denote by $\mathrm{Hom}(U_*, V_*)$ the set of all homotopisms $f_*\colon U_* \to V_*$.

The class of bimaps with homotopisms forms the *homotopism category*. There are various natural morphisms on classes of bimaps, such as adjoint-morphisms [22], so we name the categories after the morphisms rather than the objects. We are interested primarily in *isotopisms*, namely homotopisms whose constituent maps are all isomorphisms. The *autotopism group* of a bimap U_* is

$$\mathrm{Aut}(U_*) = \mathrm{Hom}(U_*, U_*) \cap \prod_{i=0}^{2} \mathrm{Aut}(U_i). \tag{2.5}$$

2.3. **Shuffling bimaps.** Since a bimap $U_* = \langle U_2, U_1, U_0, \circ\rangle$ consists of multiple components, it is tedious, both in proofs and in algorithms, to specify individual components. Accordingly, we often "shuffle" indices in our bimaps. To do this, however, we must ensure that autotopisms are unaffected by the process, so we now define precisely what we mean by shuffling. Related categorical subtleties are considered in [22]; observe that our treatment differs from that of [14].

Given a bimap $U_* = \langle U_2, U_1, U_0, \circ\rangle$ and a permutation $\sigma \in \mathrm{Sym}(\{2,1,0\})$, we define a new bimap U_*^σ. It suffices to consider just transpositions, as these generate all possible permutations, but the definition depends on the particular transposition. First consider $\sigma = (2,1)$, the transposition swapping 2 and 1. Set $U_2^\sigma := U_1$, $U_1^\sigma := U_2$, $U_0^\sigma := U_0$, and define $\circ^\sigma \colon U_2^\sigma \times U_1^\sigma \rightarrowtail U_0^\sigma$ as follows: given $u_2 \in U_2^\sigma$ and $u_1 \in U_1^\sigma$, set

$$u_2 \circ^\sigma u_1 := u_1 \circ u_2.$$

We define U_*^σ to be $\langle U_2^\sigma, U_1^\sigma, U_0^\sigma, \circ^\sigma\rangle$. A homotopism $f_* \colon U_* \to V_*$ is sent to $f_*^\sigma \colon U_*^\sigma \to V_*^\sigma$ where $f_2^\sigma := f_1$, $f_1^\sigma := f_2$, $f_0^\sigma := f_0$. In the literature this is also called the *transpose* of a bimap.

For $\sigma = (1,0)$ and $\sigma = (2,0)$ we need a slightly more elaborate construction. We consider just $(1,0)$ since $(2,0)$ works in the same way. Set $U_2^\sigma := U_2$, $U_1^\sigma := U_0^\dagger$, $U_0^\sigma := U_1^\dagger$. We switch to Greek letters when working with elements in a dual vector space. Define $\circ^\sigma : U_2^\sigma \times U_1^\sigma \rightarrowtail U_0^\sigma$ as follows: given $u_2 \in U_2^\sigma$ and $\nu_1 \in U_1^\sigma$ (so $\nu_1 \colon U_{1\sigma} \to K$), define $(u_2 \circ^\sigma \nu_1) = \nu_0 \in U_0^\sigma = \mathrm{Hom}(U_1, K)$ as follows:

$$\nu_0 \colon u_{0\sigma} \mapsto (u_2 \circ u_{0\sigma})\nu_1.$$

A complication arises with homotopisms, since interchanging f_1 and f_0 is no longer meaningful. Instead, we put $f_2^\sigma := f_2$, $f_1^\sigma := f_0^\dagger$ and $f_0^\sigma := f_1^\dagger$. In particular, f_*^σ is not a homotopism but satisfies the following condition:

$$(\forall u_2)(\forall \nu_1) \qquad (u_2 \in U_2^\sigma) \wedge (\nu_1 \in V_1^\sigma) \Rightarrow u_2 f_2^\sigma \circ^\sigma \nu_1 = (u_2 \circ^\sigma \nu_1 f_1^\sigma) f_0^\sigma.$$

This illustrates the inherent delicacy of the otherwise trivial idea of re-indexing variables. In the special case where f_* is an isotopism we observe that $\hat{f}_*^\sigma = (f_2, (f_0^\dagger)^{-1}, (f_1^\dagger)^{-1})$ is an isotopism $U_*^\sigma \to V_*^\sigma$.

2.4. **Weakly Hermitian bimaps.** A bimap is *weakly Hermitian* if it is isotopic to its transpose, namely to its $\sigma = (2,1)$ shuffle. The familiar symmetric and alternating bimaps are examples of weakly Hermitian bimaps, but the notion is substantially more general. Associated to weakly Hermitian bimaps are group invariants that respect their symmetry. If we fix an isotopism $g_* : U_* \to U_*^\dagger$, then

$$\Psi\mathrm{Isom}(U_*) = \{f_* \in \mathrm{Aut}(U_*) \colon f_* g_* = g_* f_*^\dagger\}$$

is the *group of pseudo-isometries* of U_* and does not depend on the choice of g_*.

2.5. **Algebras that operate on bimaps.** Consider the following algebras determined by a bimap U_*.

$$\mathcal{L}(U_*) = \{(f^\dagger, h^\dagger) \in \mathrm{End}(U_2) \times \mathrm{End}(U_0) \colon (f^\dagger u_2) \circ u_1 = h^\dagger(u_2 \circ u_1)\}, \tag{2.6}$$

$$\mathcal{M}(U_*) = \{(f, g^\dagger) \in \mathrm{End}(U_2) \times \mathrm{End}(U_1) \colon (u_2 f) \circ u_1 = u_2 \circ (g^\dagger u_1)\}, \tag{2.7}$$

$$\mathcal{R}(U_*) = \{(g, h) \in \mathrm{End}(U_1) \times \mathrm{End}(U_0) \colon u_2 \circ (u_1 g) = (u_2 \circ u_1)h\}, \text{ and} \tag{2.8}$$

$$\mathcal{T}(U_*) = \mathcal{L}(U_*) \oplus \mathcal{M}(U_*) \oplus \mathcal{R}(U_*). \tag{2.9}$$

Each U_i is a natural module under each of these rings, and is thus a $\mathcal{T}(U_*)$-module. Although the action is non-unital—for example, the representation of $\mathcal{L}(U_*)$ on U_1 is trivial—it is more convenient to think of each of U_2, U_1, U_0 as a module over a common ring than continually to clarify that the action on U_1 is by $\mathcal{T}(U_*)/\mathcal{L}(U_*)$, and so on. We also use the following algebra:

$$\mathcal{C}(U_*) = \left\{ f_* \in \prod_{i=0}^{2} \mathrm{End}(U_i) : (u_2 f_2) \circ u_1 = u_2 \circ (u_1 f_1) = (u_2 \circ u_1) f_0 \right\}. \tag{2.10}$$

We need one final notion. Fix bimaps U_* and V_*. As defined in [22], an *adjoint-morphism* $(f, g)\colon U_* \to V_*$ is a pair of maps $f\colon U_2 \to V_2$ and $g\colon V_1 \to U_1$ satisfying

$$(\forall u_2)(\forall v_1) \qquad (u_2 \in U_2) \wedge (v_1 \in V_1) \Rightarrow (u_2 f) \circ v_1 = u_2 \circ (v_1 g).$$

The set of all such pairs (f, g) is denoted $\mathrm{Adj}(U_*, V_*)$. This defines another category on bilinear maps distinct from those already discussed: it is an abelian category and plays a role similar to modules of rings [22, Theorem 2.27]. In particular, observe that $\mathrm{Adj}(U_*, U_*)$ is simply the ring $\mathcal{M}(U_*)$. As adjoint-bimap categories are *not* equivalent to module categories [22, Theorem 2.10], however, we must adapt some established results in module theory to suit our purpose.

3. Testing isotopism of bimaps

In this section we consider the problem of deciding if two bimaps are equivalent under isotopisms. This is an essential step in our isomorphism test for graded algebras, but it is also a problem of independent interest. We require an efficient solution to the following problem.

IsotopismCoset

Given: K-bilinear maps U_* and V_*.
Return: the coset $\mathrm{Iso}(U_*, V_*)$ of isotopisms $U_* \to V_*$.

Here, we present a basic algorithm to solve this problem; in Section 7 we introduce heuristics to speed up the construction.

Note, if $f_*\colon U_* \to V_*$ is an arbitrary isotopism, then $\mathrm{Iso}(U_*, V_*) = \mathrm{Aut}(U_*) f_*$, so the output can be encoded compactly using generators for $\mathrm{Aut}(U_*)$. If $U_* = V_*$, then the output is simply $\mathrm{Aut}(U_*)$.

3.1. **Principal autotopisms.** The difficulty of IsotopismCoset stems from having to find solutions to quadratic polynomials in multiple variables: namely, we solve for (f_2, f_1, f_0) where the parameters f_2 and f_1 occur in a product $u_2 f_2 \circ u_1 f_1$. Quadratic varieties are as complex as arbitrary varieties, but fixing any one of the f_i renders the problem substantially more tractable. Thus, we consider first the following restricted version of the autotopism group problem; in Section 3.2 we handle the coset version.

PrincipalAutotopismGroup

Given: a K-bilinear map U_* and $i \in \{2, 1, 0\}$.
Return: generators for $\mathrm{Aut}(U_*)^{(i)} := \{ f_* \in \mathrm{Aut}(U_*)\colon f_i = 1_{U_i} \}$.

The following observation leads to an efficient solution to this problem.

Proposition 3.1. *For a bimap U_*, the following hold:*

(i) $\mathrm{Aut}(U_*)^{(2)} \cong \mathcal{R}(U_*)^\times$*;*
(ii) $\mathrm{Aut}(U_*)^{(1)} \cong \mathcal{L}(U_*)^\times$*; and*
(iii) $\mathrm{Aut}(U_*)^{(0)} \cong \mathcal{M}(U_*)^\times$*.*

Proof. Following our discussion in Section 2.3, we can assume $i = 0$ after a possible shuffling of the variables. (We stress once more that reindexing requires some readjustment of the resulting isotopisms.) If $f_* \in \mathrm{Aut}(U_*)$ and $f_0 = 1$, then

$$(u_2 f_2) \circ u_1 = (u_2 f_2) \circ (u_1 f_1^{-1} f_1) = (u_2 \circ (u_1 f_1^{-1})) f_0 = u_2 \circ (u_1 f_1^{-1}),$$

so $(f_2, f_1^{-1}) \in \mathcal{M}(U_*)$. If $(f_2, f_1) \in \mathcal{M}(U_*)$ then $(f_2, f_1^{-1}, 1_{U_0}) \in \mathrm{Aut}(U_*)$. □

Algorithm 1 Principal Autotopism Group

Input: a K-bilinear map U_* and $i \in \{2, 1, 0\}$.
Output: generators for $\mathrm{Aut}(U_*)^{(i)}$.

1: Choose a permutation σ on $\{2, 1, 0\}$ with $i\sigma = 0$.
2: Solve a system of linear equations to find a basis for $\mathcal{M}(U_*^\sigma)$.
3: Use [6, Theorem 2.3] to compute generators X for the group of units of $\mathcal{M}(U_*^\sigma)$.
4: Set $G = \langle (f, g^{-1}, 1)^\sigma : (f, g) \in X \rangle \leqslant \prod_{i=0}^{2} \mathrm{Aut}(U_i)$.
5: **return** G.

Proposition 3.2. *Algorithm* 1 *solves* PrincipalAutotopismGroup*. As a deterministic algorithm it runs in time* $O((\dim U_*)^{2\omega} \log^2 |K| + \mathrm{char}\ K)$*; a Las Vegas variant runs in time* $O((\dim U_*)^{2\omega} \log^2 |K|)$.

Proof. Since the correctness of the algorithm is clear from Proposition 3.1 and the mechanics of shuffling variables, we focus on the complexity. Line 2 involves solving a system of $(\dim U_2)(\dim U_1)(\dim U_0)$ linear equations in $(\dim U_2)^2 + (\dim U_1)^2$ variables, which can be done in time $O((\dim U_*)^{2\omega} \log^2 |K|)$. Line 3 invokes the algorithm of [6, Theorem 2.3], which depends on the ability to factor polynomials over K. The algorithm runs in Las Vegas polynomial-time $O((\dim A)^{2\omega} \log^2 |K|)$ if we use Las Vegas polynomial factorization routines such as that of [9]. A deterministic algorithm is known when the ground field of K can be listed: in this case Line 3 runs in time $O((\dim A)^{2\omega} \log^2 |K| + \mathrm{char}\ K)$. The remaining steps of Algorithm 1 have negligible influence on the timing, so the result follows. □

3.2. **Extending to isotopisms.** Our next objective is to solve a single instance of isotopism. We focus first on principal isotopisms, and assume that $i = 0$ by shuffling coordinates.

PrincipalIsotopism

Given: K-bilinear maps U_* and V_* and a map $f_i \colon U_i \to V_i$ for fixed $i \in \{2, 1, 0\}$.
Return: an isotopism $f_* \colon U_* \to V_*$ extending f_i.

Just as the construction of the principal autotopism group is a problem in rings, the construction of a principal isotopism resembles a problem in modules. As we indicated in Section 2, however, it is not precisely a module problem that we solve.

Definition 3.3. *An* orthogonal decomposition *of a bimap U_* is a pair of direct decompositions $U_2 = \bigoplus_j U_{2j}$ and $U_1 = \bigoplus_k U_{1k}$ such that $U_{2j} \circ U_{1k} = 0$ if $j \neq k$. Each U_{ij} is an* orthogonal factor.

For example, if $U_* = \langle K^2, K^3, K, \circ \rangle$ where

$$u_2 \circ u_1 = u_2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} u_1^\dagger$$

then $U_2 = K(1,0) \oplus K(0,1)$ and $U_1 = K(1,0,0) \oplus K(0,1,0) \oplus K(0,0,1)$ is an orthogonal decomposition. More generally, in terms of structure constants this implies that $A^{(k)} = [a_{ij}^k]$ is block diagonal, the blocks coinciding with the (U_{2j}, U_{1j}) pairs; see [22, Section 2.4] for details.

Our plan is to imitate the algorithm of [4], which builds a module isomorphism one direct summand at a time. Both that algorithm and our adaptation rely on the following construction. If $X \subseteq \mathrm{End}_K(V)$, then let $\overline{X}$ be the semigroup generated by X and let $K\langle X \rangle$ be the K-linear span of $\overline{X}$. (Contrary to the usual notion of enveloping algebra, $K\langle X \rangle$ need not be unital: this occurs if, and only if, the identity can be written as a linear combination of elements of $\overline{X}$.)

Theorem 3.4 ([4, Corollary 2.5])**.** *There is a polynomial-time algorithm that, given $X \subseteq \mathrm{End}_K(V)$, decides if $K\langle X \rangle$ is nilpotent and, if not, returns a product of elements in $\overline{X}$ that is not nilpotent.*

The algorithm in [4] uses non-nilpotent elements to decompose the modules into direct summands. Instead of module isomorphisms we construct principal isotopisms; instead of direct summands we use orthogonal factors. We capture the key recursive step with the following technical definition.

Definition 3.5. *Fix $f_0 \colon U_0 \to V_0$. A* partial f_0-isotopism *of bimaps U_* and V_* is an isotopism $g_* = (g_2, g_1, g_0)$ defined on the restriction to some orthogonal factors of U_* and V_*, and such that $g_0 = f_0$. A partial f_0-isotopism is* maximal *if it is not a restriction to proper subspaces of another partial f_0-isotopism.*

The idea is to build a (possibly nilpotent) ring from two sets of adjoint-morphisms. If this ring contains an invertible element, then we find the desired principal isotopism. Following [4], we propose Algorithm 2 to construct a maximal partial isotopism.

Algorithm 2 Partial Principal Isotopism

Input: bimaps U_* and V_*, and an isomorphism $f_0 \colon U_0 \to V_0$.
Output: a maximal partial f_0-isotopism.

1: $\mathcal{X} \leftarrow \mathrm{Basis}(\mathrm{Adj}(U_*, V_*^{f_0}))$; $\mathcal{Y} \leftarrow \mathrm{Basis}(\mathrm{Adj}(V_*^{f_0}, U_*))$.
2: $A \leftarrow K\langle xy : x \in \mathcal{X}, y \in \mathcal{Y} \rangle \subset \mathrm{End}(U_2) \times \mathrm{End}(U_1)^{\mathrm{op}}$.
3: For $i = 1, 2$, $U_i^- \leftarrow U_i$; $U_i^+ \leftarrow 0$; $f_i \leftarrow 0$.
4: **while** A has $z = xyw$ not nilpotent with $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ **do**
5: Find $n \geqslant 0$ such that, for $i = 1, 2$, $U_i^- = \ker z^n \oplus \mathrm{im}\, z^n$.
6: $U_i^- \leftarrow \ker z^n$; $U_i^+ \leftarrow U_i^+ \oplus \mathrm{im}\, z^n$; $f_i \leftarrow f_i \oplus \mathrm{res}_{\mathrm{im}\, z^n}(x)$.
7: Restrict A to $\ker z^n$.
8: **return** $f_* = (f_2 \colon U_2^+ \to V_2^+, f_1 \colon U_1^+ \to V_1^+, f_0 \colon U_0 \to V_0)$.

Proposition 3.6. *Algorithm* 2 *is deterministic and constructs a maximal f_0-isotopism in polynomial time $O((\dim U_*)^{2\omega} \log^2 |K|)$.*

Proof. The objective of the algorithm is to find an invertible element of $\mathrm{Adj}(U_*, V_*^{f_0})$. To do this, we first create a (possibly non-unital) algebra A in Line 2 by composing the sets of homomorphisms created in Line 1. Observe that composition in the second variable is in the op-ring $\mathrm{End}(U_1)^{\mathrm{op}}$.

First, note that $A \subset \mathcal{M}(U_*)$. Secondly, if f_0 extends to an isotopism $f_*\colon U_* \to V_*$ then $(f_2, f_1^{-1}) \in \mathrm{Adj}(U_*, V_*^{f_0})$ and $(f_1, f_2^{-1}) \in \mathrm{Adj}(V_*^{f_0}, U_*)$; in particular, A contains units. However, finding a unit of A does not guarantee that we can extract an invertible element of $\mathrm{Adj}(U_*, V_*^{f_0})$.

We claim that the loop starting in Line 4 maintains the following invariants: for $i \in \{2, 1\}$, $U_i = U_i^+ \oplus U_i^-$ is an orthogonal decomposition of U_i; and (f_2, f_1, f_0) is a partial f_0-isotopism. Clearly, this is true at the start.

By its construction in Line 4, clearly $z \in \mathcal{M}(U_*)$, and so $z^n \in \mathcal{M}(U_*)$. Note, for every $b \in \mathcal{M}(U_*)$, the decomposition $U_i = \ker b \oplus \operatorname{im} b$ is orthogonal. Therefore we maintain throughout an orthogonal decomposition $U_i = U_i^+ \oplus U_i^-$. Furthermore, by Fitting's lemma, z^n is invertible on $\operatorname{im} z^n$. The guard of the loop in Line 4 is a call to Theorem 3.4, which provides $x \in \mathrm{Adj}(U_*, V_*^{f_0})$ such that $z = xyw$. As z^n is invertible on $\operatorname{im} z^n$, and $z^n = x \cdots$, it follows that x is injective on $\operatorname{im} z^n$. Since all spaces are finite, this injection is a bijection. Therefore the extension of f_i by the restriction of x to the image of z^n remains a partial f_0-isotopism, as required.

Finally, the loop continues while A contains a non-nilpotent element. Thus, the partial f_0-isotopism is maximal and the output is correct.

The major work is solving the system of linear equations in Line 1; this results in the complexity stated in the theorem. □

Remark. In many settings, invertible elements of $\mathrm{Adj}(U_*, V_*^{f_0})$ may be found by random search with high probability, but there are examples that require an exponential number of samples to return an invertible element. Nevertheless, once $\mathrm{Adj}(U_*, V_*^{f_0})$ is constructed, it is sensible to test a small number of random elements.

The following is now immediate.

Theorem 3.7. *There is a deterministic, polynomial-time algorithm to solve* PrincipalIsotopism.

We are finally ready to present Algorithm 3, the main result of this section.

Viewing the group G in Line 4 of this algorithm as a parameter, the following is an immediate consequence of the results and algorithms of this section.

Theorem 3.8. *Algorithm* 3 *solves* IsotopismCoset *and runs deterministically in time $O(|G| \max_i\{(\dim U_i)^{2\omega}\} \log^2 |K|)$, where G is the group in Line* 4.

4. Testing isomorphism of graded algebras

Our algorithm to decide isomorphism between graded algebras proceeds under the assumption that an isomorphism exists. If this does not occur then the test is aborted. A standard mechanism to do this is to raise an exception. This means that all further steps are aborted and the algorithm backtracks to the nearest place that can handle the exception. Recall from Proposition 2.3 that we assume an algebra A_* is graded by a monoid M that has size polynomial in $\dim A_*$.

Algorithm 3 Isotopism coset

Input: bimaps U_* and V_*.
Output: the coset $\mathrm{Iso}(U_*, V_*)$ of isotopisms $U_* \to V_*$.

1: Choose a permutation σ on $\{2, 1, 0\}$ with $\dim U_{0\sigma}$ minimized.
2: **if** $\dim U_{0\sigma} \neq \dim V_{0\sigma}$ **then return** $\emptyset$.
3: Choose any isomorphism $f_{0\sigma} : U_{0\sigma} \to V_{0\sigma}$.
4: Choose G such that $\mathrm{Aut}(U_*)|_{U_{0\sigma}} \leqslant G \leqslant \mathrm{Aut}(U_{0\sigma})$. /* *see Section* 7 */
5: $I \leftarrow \emptyset$.
6: **for all** $g \in G$ **do**
7: /* *Algorithms* 1 *&* 2 */
8: Find the coset C of isotopisms $h_* \colon U_* \to V_*$ with $h_{0\sigma} = g f_{0\sigma}$.
9: $I \leftarrow I \cup C$.
10: **return** I

4.1. **Extending isotopisms to graded isomorphisms.** The isomorphism algorithm proceeds by attempting to extend isotopisms between the bimaps obtained by restricting the given products to certain fixed homogeneous components. We therefore begin by considering the necessary extension problem; our solution is summarized in Algorithm 4. As A_* and B_* are generated in degrees T, we assume that we have a homotopism whose restriction to T, namely $f_T = \{(f_t \colon A_t \to B_t) \colon t \in T\}$, is defined. Our task is to extend f_T to a graded algebra homomorphism $f_* \colon A_* \to B_*$.

For $s \in M \setminus T$, define

$$A_{\otimes s} \;:=\; \bigoplus_{\substack{s = s_1 + s_2,\\ s_i \notin \{0, s\}}} A_{s_1} \otimes A_{s_2} \;\subset\; A_* \otimes A_*.$$

Given $s \in M$, we can construct $\{(s_1, s_2) \in M \times M : s = s_1 + s_2\}$ in at most $|M|^2 \leqslant (\dim A_*)^2$ steps. Define

$$f_{\otimes s} \;:=\; \bigoplus_{\substack{s = s_1 + s_2,\\ s_i \notin \{0, s\}}} f_{s_1} \otimes f_{s_2} \in \mathrm{Hom}(A_{\otimes s}, B_{\otimes s}),$$

where $f_{s_1} \otimes f_{s_2}$ is defined component-wise on $A_{s_1} \otimes A_{s_2}$. For $s \in M$, let $\prec s = \{u \in M : u \prec s\}$. For $R \subset M$, write $R \prec s$ if $r \prec s$ for every $r \in R$. If A_* is generated in degrees T, then setting $s = \sum_{t \in T} t$ implies that $T \prec s$.

Proposition 4.1. *Algorithm* 4 *is correct and runs deterministically in time*

$$O(|M|(\dim A_*)^{2\omega} \log^2 |K|).$$

Proof. Consider first the correctness of the algorithm. Observe that $f_* \colon A_* \to B_*$ is a graded homomorphism if, and only if, for all s and t, (f_s, f_t, f_{s+t}) is a homotopism from the bimap $A_s \times A_t \rightarrowtail A_{s+t}$ to $B_s \times B_t \rightarrowtail B_{s+t}$.

We claim that the loop starting on Line 2 has the following invariants:

(i) R is an interval closed set of indices;
(ii) for all $r \in R$, the map $f_r \colon A_r \to B_r$ is defined; and
(iii) for all $r, r' \in R$, if $r + r' \in R$, then $(f_r, f_{r'}, f_{r+r'})$ is an isotopism from $A_r \times A_{r'} \rightarrowtail A_{r+r'}$ to $B_r \times B_{r'} \rightarrowtail B_{r+r'}$.

The loop terminates when $R = M$, so $f_* \colon A_* \to B_*$ is defined and is consequently a graded algebra isomorphism.

Algorithm 4 Extending Homotopisms

Input: finite M-graded K-algebras A_* and B_* generated in degrees $T \subset M$, and $f_T = \{(f_t \colon A_t \to B_t) \colon t \in T\}$ such that for every $t, r \in T$ with $t + r \in T$, the triple (f_t, f_r, f_{t+r}) is a homotopism from $A_r \times A_t \rightarrowtail A_{r+t}$ to $B_r \times B_t \rightarrowtail B_{r+t}$.

Output: an algebra homomorphism $f_* \colon A_* \to B_*$ extending f_T, or raise an exception if f_T does not extend to an algebra homomorphism.

1: $R \leftarrow T$.
2: **while** $R \neq M$ **do**
3: Choose $s \in M \setminus R$ such that $\{r \in M \colon r \prec s, r \neq s\} \subset R$.
4: Induce $\pi_{s,A_*} \colon A_{\otimes s} \to A_s$ and likewise $\pi_{s,B_*} \colon B_{\otimes s} \to B_s$.
5: Compute $\ker \pi_{s,A_*}$ and $\ker \pi_{s,B_*}$.
6: **if** $(\ker \pi_{s,A_*}) f_{\otimes s} \leqslant \ker \pi_{s,B_*}$ **then**
7: $f_s \leftarrow \pi_{s,A_*}^{-1} \cdot f_{\otimes s} \cdot \pi_{s,B_*} \in \operatorname{Hom}(A_s, B_s)$.
8: $R \leftarrow R \cup \{s\}$.
9: **else**
10: **raise exception** no extension exists at s.
11: **return** f_*.

Consider any s selected in Line 3. Since A_* is generated in degrees T, and $T \subset R$, it follows that π_{s,A_*} is surjective and

$$A_s \cong A_{\otimes s} / \ker \pi_{s,A_*}.$$

The same holds for B_*. By our choice of s, for every $s_1, s_2 \notin \{0, s\}$ with $s = s_1 + s_2$, f_{s_1} and f_{s_2} are defined, so $f_{\otimes s}$ is defined. If $(\ker \pi_{s,A_*}) f_{\otimes s} \leqslant \ker \pi_{s,B_*}$, then we may induce $f_s \colon A_s \to B_s$ on the generators of A_s as follows: for $a_{s_1} \in A_{s_1}$ and $a_{s_2} \in A_{s_2}$,

$$(a_{s_1} \circ a_{s_2}) f_s := (a_{s_1} \otimes a_{s_2}) f_{\otimes s} \equiv a_{s_1} f_{s_1} \circ a_{s_2} f_{s_2} \bmod \ker \pi_{s,B_*}.$$

Conversely, if $f_{\prec s}$ extends to a graded isomorphism $f_* \colon A_* \to B_*$, then f_s is defined as above. Thus, if $(\ker \pi_{s,A_*}) f_{\otimes s} \not\leqslant \ker \pi_{s,B_*}$ we conclude that $\{f_t \colon t \in T\}$ does not extend, and raise an exception to abort all subsequent steps.

Next, we analyze the timing. The loop executes at most $|M| \in (\dim A_*)^{O(1)}$ iterations, and the timing in each one is dominated by the computation of $\ker \pi_{s,A_*}$ and $\ker \pi_{s,B_*}$ and the subsequent membership test in the latter. Each requires solving systems of linear equations in $(\dim A_s)^2$ variables. In total this takes $O(\sum_s (\dim A_s)^{2\omega} \log^2 |K|)$ steps, as stated. □

4.2. **The isomorphism test.** For an M-graded algebra A_* and $S \subset M$, define $A_S = \bigoplus_{s \in S} A_s$; for a graded homomorphism f_*, define $f_S = \bigoplus_{s \in S} f_s$. Algorithm 5 is our isomorphism test for graded algebras. The mechanism for selecting S in Line 1 is discussed in Section 4.3.

Proposition 4.2. *Algorithm* 5 *is correct.*

Proof. If $f_* \colon A_* \to B_*$ is a graded isomorphism, then the restriction

$$\left(\bigoplus_{t \in T} f_t, \bigoplus_{s \in S} f_s, \bigoplus_{s \in S, t \in T} f_{s+t} \right)$$

Algorithm 5 Graded Isomorphism Coset

Input: finite M-graded algebras A_* and B_* generated in degrees $T \subset M$.
Output: the coset of graded isomorphisms $A_* \to B_*$, or $\emptyset$ if $A \ncong B$.
1: Choose $\emptyset \neq S \subset M$. /* *see Section* 4.3 */
2: Restrict multiplication in A_* to obtain the bimap $A_T \times A_S \rightarrowtail A_{S+T}$.
3: Similarly, obtain the bimap $B_T \times B_S \rightarrowtail B_{S+T}$.
4: $I \leftarrow \mathrm{Iso}(A_T \times A_S \rightarrowtail A_{S+T}, B_T \times B_S \rightarrowtail B_{S+T})$, using Algorithm 3.
5: $\Gamma \leftarrow \{f_* : A_* \to B_* : f_* \text{ extends some } (f_T, f_S, f_{S+T}) \in I\}$, using Algorithm 4.
6: **return** Γ.

is an isotopism from $A_T \times A_S \rightarrowtail A_{S+T}$ to $B_T \times B_S \rightarrowtail B_{S+T}$. Since A_* is generated in degrees T, for each $s \in M \setminus T$, f_s is determined by $f_T = \bigoplus_{t\in T} f_t$. In particular, $f_* = 1$ if, and only if, $f_T = 1$. Hence, the mapping

$$f_* \mapsto \left(\bigoplus_{t\in T} f_t, \bigoplus_{s\in S} f_s, \bigoplus_{s\in S, t\in T} f_{s+t} \right)$$

is injective. The algorithm constructs the inverse image of this injection, and so returns the coset of graded isomorphisms $A_* \to B_*$. □

Proof of Theorem 1.1. We analyze the complexity of Algorithm 5. If the algorithm discovers that $A_* \ncong B_*$, then it terminates. Hence, the case that dominates complexity is $A_* \cong B_*$. Using Algorithm 3, $I = \mathrm{Iso}(U_*, V_*)$ is constructed in time

$$O(\min\{|\operatorname{Aut}(A_T)|, |\operatorname{Aut}(A_S), |\operatorname{Aut}(A_{S+T})|\}(\dim A_*)^{2\omega} \log^2 |K|),$$

and hence in time $O(|\operatorname{Aut}(A_{S+T})|(\dim A_*)^{2\omega} \log^2 |K|)$. The remaining time to construct $\mathrm{Iso}(A_*, B_*)$ is $O(|I|)$. Observe that

$$|I| = |\operatorname{Aut}(U_*)| \leqslant |\mathcal{M}(U_*)^\times| \cdot |\operatorname{Aut}(A_{S+T})|.$$

The complexity stated in Theorem 1.1 now follows by substituting $T = \{1\}$ and $S = \{s\}$, where s is the largest positive integer such that $A_{s+1} \neq 0$. □

4.3. **Selecting optimal indices.** We now discuss the issue left open in Line 1 of Algorithm 5: how to choose the subset, S, of optimal indices. Our aim is to predict the order of

$$\operatorname{Aut}(A_T \times A_S \rightarrowtail A_{S+T})$$

without computing it. This allows us in Line 1 of Algorithm 5 to sample several subsets S to find a selection whose estimated work is either minimal, or below an acceptable threshold. If U_* is the bimap $A_T \times A_S \rightarrowtail A_{S+T}$, then by definition there is an exact sequence

$$1 \to \mathcal{M}(U_*)^\times \to \operatorname{Aut}(U_*) \to \operatorname{Aut}(U_0), \tag{4.3}$$

from which we immediately obtain the bound

$$|\operatorname{Aut}(U_*)| \leqslant |\mathcal{M}(U_*)^\times| \cdot q^{(\dim U_0)^2}. \tag{4.4}$$

This bound suffices to prove our main theorems, but more precise bounds on $|\operatorname{Aut}(U_*)|$ can be obtained with additional work. We include a brief discussion here both because our implementation uses the better bounds, and also because future analyses of the complexity of our algorithm for specific families of inputs may require them.

In [7, Theorem 3.2] a property of autotopisms is given which leads to a general bound on $|\mathrm{Aut}(U_*)|$. However, a better bound using the rings defined in Section 2.5 may be derived from the exact sequences in [23, Theorem 1.2]. Let $\mathrm{Out}(R)$ be the group of outer automorphisms of a ring R. If R is equipped with an involution $a \mapsto \bar{a}$, then $R^\# = \{a \in R \colon a\bar{a} = 1\}$ denotes its group of unitary elements, and $\mathrm{Out}^\#(R)$ is the subgroup of $\mathrm{Out}(R)$ commuting with the involution.

Proposition 4.5. *For a bimap $U_* = (U_2, U_1, U_0, \circ)$ the following holds:*

$$\begin{aligned}\frac{|\mathcal{T}(U_*)^\times|}{|\mathcal{C}(U_*)^\times|} &\leqslant |\mathrm{Aut}(U_*)| \\ &\leqslant |\mathcal{T}(U_*)^\times| \cdot |\mathrm{Out}(\mathcal{T}(U_*))| \cdot \min\{|\mathrm{Aut}_{\mathcal{T}(U_*)}(U_i)| : 0 \leqslant i \leqslant 2\}.\end{aligned}$$

If U_* is weakly Hermitian then $\mathcal{T}(U_*)$ and $\mathcal{C}(U_*)$ are rings with involutions, and

$$\begin{aligned}\frac{|\mathcal{T}(U_*)^\#|}{|\mathcal{C}(U_*)^\#|} &\leqslant |\Psi\mathrm{Isom}(U_*)| \\ &\leqslant |\mathcal{T}(U_*)^\#| \cdot |\mathrm{Out}^\#(\mathcal{T}(U_*))| \cdot \min\{|\mathrm{Aut}_{\mathcal{T}(U_*)}(U_i)| : 0 \leqslant i \leqslant 2\}.\end{aligned}$$

Each bound can be computed in time polynomial in $\sum_{i=0}^{2} \dim U_i$.

5. Proof of Theorem 1.2

The efficiency of our test for isomorphism between graded algebras A_* and B_* depends critically on two conditions: first, we can find a homogeneous component A_{S+T} of moderate size; secondly, the order of $\mathrm{Aut}(A_T \times A_S \rightarrowtail A_{S+T})$ is manageable. In this section, we consider a natural family of nilpotent Lie algebras whose basic parameters illustrate the performance of our isomorphism test. In doing so we prove Theorem 1.2.

Every nilpotent matrix Lie algebra $\mathfrak{L}_* \leqslant \mathfrak{gl}(V)$ has a nontrivial 0-eigenspace V_1. Recursively, for $i \geqslant 1$, let $V_{i+1} \leqslant V$ so that V_{i+1}/V_i is the 0-eigenspace of the representation of $\mathfrak{L}_*$ on V/V_i. The flag $0 < V_1 < \ldots < V_\ell = V$ is denoted $\mathcal{F}(\mathfrak{L}_*)$. Conversely, associated to each flag $\mathcal{F}$ of V is a unique maximal nilpotent Lie subalgebra $\mathfrak{P}(\mathcal{F})_* \leqslant \mathfrak{gl}(V)$ such that $\mathcal{F}(\mathfrak{P}(\mathcal{F})_*) = \mathcal{F}$. We say $\mathfrak{L}_*$ is *dense* if

$$[\mathfrak{L}_*, \mathfrak{L}_*] = [\mathfrak{P}(\mathcal{F}(\mathfrak{L}_*))_*, \mathfrak{P}(\mathcal{F}(\mathfrak{L}_*))_*].$$

5.1. **Comparing results.** Before presenting the proof of Theorem 1.2, we pause to compare its complexity to that of other algorithms for algebra isomorphism, and to other well known computational problems. It is helpful to use *L-notation* which we define in terms of logarithms to base $q = |K|$:

$$L_n[\alpha, c] = q^{(c+o(1))(\log n)^\alpha (\log\log n)^{\delta(0,\alpha)}}, \qquad \text{where } \delta(0,\alpha) = \begin{cases} 0 & \alpha = 0 \\ 1 & \alpha > 0 \end{cases}.$$

This function interpolates between polylogarithms $L_n[0, c] \in \tilde{O}((\log n)^c)$, polynomials $L_n[1, c] \in \tilde{O}(n^c)$, quasi-polynomials $L_n[2, c] \in \tilde{O}(n^{c \ln n})$, and so forth. For context, the heuristic estimates (in base e) for the cost to factor an integer n are $L_n[\frac{1}{2}, 1]$ for the quadratic sieve, and $L_n[\frac{1}{3}, \sqrt[3]{64/9}]$ for the number field sieve.

Assuming the size of the field is constant, the complexity of graded algebra isomorphism in Theorem 1.1 is $L_n[2, \frac{1}{2}]$.

We compare this to the general isomorphism test for Lie algebras presented in [11]. That algorithm must select the first homogeneous component $\mathfrak{L}_1$. In the worst case—where $\mathfrak{L}_* = \mathfrak{L}_1 \oplus \mathfrak{L}_2$ and $\dim \mathfrak{L}_1 = r \dim \mathfrak{L}_*$ for some constant $r \leqslant 1$—it exhaustively searches $\mathrm{Aut}(\mathfrak{L}_1)$, resulting in a complexity of $L_n[2,1]$. For cases in which $\dim \mathfrak{L}_1 = r\sqrt{\dim \mathfrak{L}_*}$, however, the complexity improves to $L_n[1,c]$. This shows how the Hilbert series $\sum_i (\dim \mathfrak{L}_i)x^i$ of the input influences the complexity of [11]; as we explain below, the Hilbert series exerts an influence over the complexity of our algorithm that is both more subtle and more emphatic.

Another recent approach to isomorphism [7, 23] can be applied to graded Lie algebras of class 2. This exploits invariant algebras of bimaps and is particularly effective when there are large automorphism groups. This method also has complexity ranging from a worst case of $L_n[2,1]$ down to nearly optimal run times of $L_n[0,2\omega]$ for inputs such as generalized Heisenberg Lie algebras.

Now, let us consider the complexity of our current algorithm as it appears in Theorem 1.2. First, ε measures the "area" occupied by the dense algebra $\mathfrak{L}_*$. The presence of a few large blocks around the middle of $\mathcal{F}(\mathfrak{L}_*)$ makes ε comparatively small, and hence slows down the performance of our algorithm. Secondly, m is the dimension of the smallest $\mathfrak{L}_{s+1}$ we encounter, subject to the product $\mathfrak{L}_1 \times \mathfrak{L}_s \rightarrowtail \mathfrak{L}_{s+1}$ being nondegenerate (the condition $s \leqslant 1 + d/2$ in the formula for m ensures this). As illustrated in Figure 1, small values of m^2/ε correspond to super block diagonal layers near the middle that are as thin as possible. The dimensions of the possible layers are determined by the Hilbert series of the input.

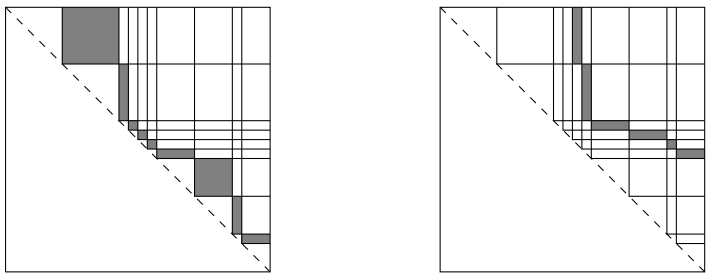

FIGURE 1. Contrasting the work to construct automorphisms

Observe that the work to construct automorphisms by lifting from a fixed shaded layer is roughly q^{a^2}, where q is the size of the field and a is the area of the shaded region (the dimension of the corresponding homogeneous component). The shaded area on the left of Figure 1 is roughly 3 times that on the right. Thus, if w is the work needed to search for automorphisms using the shaded region on the left, then using that on the right decreases the work to $w^{1/9}$.

5.2. **Multiplication tables.** We work with $\mathfrak{L}_*$ as matrices relative to a basis exhibiting $\mathcal{F}(\mathfrak{L}_*)$, so elements of $\mathfrak{L}_*$ are block upper-triangular matrices with 0's on the diagonal. Although the specific entries in each block determine the structure of the algebra, much can be deduced simply by studying the block structure. For $x \in \mathfrak{L}_*$, denote by x_{st} the block of x in (block) row s and (block) column t. For

$x, y \in \mathfrak{L}_*$, the following is the general formula for the product on $\mathfrak{L}_*$:

$$\left[\sum_{1\leqslant s<t\leqslant d} x_{st},\ \sum_{1\leqslant u<v\leqslant d} y_{uv}\right] = \sum_{1\leqslant s<u<t\leqslant d} x_{su}y_{ut} - y_{su}x_{ut}. \tag{5.1}$$

Each product $x_{su}y_{ut} - y_{su}x_{ut}$ may be specified by structure constants depending solely on (s,t); we demonstrate this below for $d = 6$. (As $\mathfrak{L}_*$ is generated in degree 1, we display only the structure constants for each product $\mathfrak{L}_1 \times \mathfrak{L}_n \rightarrowtail \mathfrak{L}_{n+1}$ for $n = 1, \dots, 5$.)

	x_{12}	x_{23}	x_{34}	x_{45}	x_{56}	x_{13}	x_{24}	x_{35}	x_{46}	x_{14}	x_{25}	x_{36}	x_{15}	x_{26}	x_{16}
x_{12}	·	a_1	·	·	·	·	b_1	·	·	·	c_1	·	·	d_1	·
x_{23}	$-a_1^\dagger$	·	a_2	·	·	·	·	b_2	·	·	·	c_2	·	·	·
x_{34}	·	$-a_2^\dagger$	·	a_3	·	$-b_1^\dagger$	·	·	b_3	·	·	·	·	·	·
x_{45}	·	·	$-a_3^\dagger$	·	a_4	·	$-b_2^\dagger$	·	·	$-c_1^\dagger$	·	·	·	·	·
x_{56}	·	·	·	$-a_4^\dagger$	·	·	·	$-b_3^\dagger$	·	·	$c_2^\dagger$	·	$-d_1^\dagger$	·	·

This example illustrates two competing tensions that determine our success. On one hand, we want to select s such that $\mathrm{Aut}(\mathfrak{L}_1 \times \mathfrak{L}_s \rightarrowtail \mathfrak{L}_{1+s})$ is small. This occurs when the corresponding product has the most constraining equations, which happens generically when s is small. On the other hand, our method to compute $\mathrm{Aut}(\mathfrak{L}_1 \times \mathfrak{L}_s \rightarrowtail \mathfrak{L}_{1+s})$ requires that we list $\mathrm{Aut}(\mathfrak{L}_{1+s})$, and the order of this group tends to decrease as s increases. This explains why the best choice of s is typically near the middle.

5.3. **Estimating $\mathrm{Aut}(\mathfrak{L}_1 \times \mathfrak{L}_s \rightarrowtail \mathfrak{L}_{1+s})$.** To prove Theorem 1.2 it suffices to consider the bound on $\mathrm{Aut}(\mathfrak{L}_1 \times \mathfrak{L}_s \rightarrowtail \mathfrak{L}_{1+s})$ arising from the exact sequence (4.3). In particular we show that if $s \leqslant 1 + d/2$, then the choice $U_* = (\mathfrak{L}_1, \mathfrak{L}_s, \mathfrak{L}_{1+s}, [,])$ of homogeneous component in Theorem 1.1 leads to the complexity stated in Theorem 1.2.

Put $U_2 := \mathfrak{L}_1$, $U_1 := \mathfrak{L}_s$, and $U_0 := \mathfrak{L}_{1+s}$. It suffices to establish the bounds

$$|\mathcal{M}(U_*)^\times| \leqslant q^{d^2} \leqslant |\mathfrak{L}|^{1/\varepsilon}, \qquad |\mathrm{Aut}(U_0)| \leqslant q^{m^2d^2} \leqslant |\mathfrak{L}|^{m^2/\varepsilon}. \tag{5.2}$$

As suggested in Section 4.3 we can derive more subtle bounds by applying Proposition 4.5, but the analysis offers little insight into the complexity of our algorithm.

We first make an observation. If $X_*^i = (X_2, X_1, X_0^i, \circ_i)$ $(i = 1, 2)$ are two bimaps on a common domain $X_2 \times X_1$, but with possibly different codomains and products, then we can define a new bimap $Y_* = X_*^1 \cap X_*^2$ as follows. Put $Y_2 := X_2$, $Y_1 := X_1$, $Y_0 := X_0^1 \oplus X_0^2$, and define $\circ\colon Y_2 \times Y_1 \rightarrowtail Y_0$ by

$$y_2 \circ y_1 := (y_2 \circ_1 y_1) \oplus (y_2 \circ_2 y_1).$$

Note that $\mathcal{M}(X_*^1 \cap X_*^2) = \mathcal{M}(X_*^1) \cap \mathcal{M}(X_*^2)$, so selecting a basis for U_0 decomposes U_* as $\bigcap_i U_*^i$, thereby making it simpler to compute the ring of adjoints.

To compute $\mathcal{M}(U_*)$ we solve equations of the form $FA = AG^\dagger$, where A is a matrix of structure constants of U_* of the form

$$A = \begin{bmatrix} \cdot & a_1 & & & \\ & \ddots & \ddots & & \\ & & \cdot & & a_\ell \\ -a_1^\dagger & \cdot & & & \\ & \ddots & \ddots & & \\ & & & -a_\ell^\dagger & \cdot \end{bmatrix}. \tag{5.3}$$

We caution that our illustration shows the structure constants for a typical configuration but changing the numbers of blocks both changes the number of boxes and alters their positions. To solve this system of equations, we decompose U_* into terms U_*^i using the block structure, as follows:

$$\begin{bmatrix} \cdot & a_1 & & \\ & \ddots & \ddots & \\ & & \cdot & a_\ell \\ -a_1^\dagger & \cdot & & \\ & \ddots & \ddots & \\ & & -a_\ell^\dagger & \cdot \end{bmatrix} = \begin{bmatrix} \cdot & a_1 & & \\ & \ddots & \ddots & \\ & & \cdot & \cdot \\ -a_1^\dagger & \cdot & & \\ & \ddots & \ddots & \\ & & \cdot & \cdot \end{bmatrix} \cap \cdots \cap \begin{bmatrix} \cdot & \cdot & & \\ & \ddots & \ddots & \\ & & \cdot & a_\ell \\ \cdot & \cdot & & \\ & \ddots & \ddots & \\ & & -a_\ell^\dagger & \cdot \end{bmatrix}.$$

This is not precisely a decomposition into the blocks U_*^i because each term on the right hand side is padded with zeros, thereby defining a degenerate extension $\hat{U}_*^i$ of the desired U_*^i. However, the relationship between $\mathcal{M}(\hat{U}_*^i)$ and $\mathcal{M}(U_i^*)$ is straightforward:

$$\mathcal{M}(\hat{U}_*^i) = \mathcal{M}(U_*^i) + \operatorname{Hom}\left(U_2^i, (U_1^i)^\perp\right) \times \operatorname{Hom}\left(U_1^i, (U_2^i)^\perp\right), \text{ and}$$

$$\mathcal{M}\left(\begin{bmatrix} \cdot & \cdot & & \\ & \cdot & a_i & \\ & & \cdot & \cdot \\ \cdot & \cdot & & \\ & -a_i^\dagger & \cdot & \\ & & \cdot & \cdot \end{bmatrix}\right) = \left\{ \left(\begin{bmatrix} * & \cdot & * & * & \cdot & * \\ \cdot & \alpha & \cdot & \cdot & \beta & \cdot \\ * & \cdot & * & * & \cdot & * \\ * & \cdot & * & * & \cdot & * \\ \cdot & \gamma & \cdot & \cdot & \delta & \cdot \\ * & \cdot & * & * & \cdot & * \end{bmatrix}, \begin{bmatrix} * & \cdot & * & \cdot & * \\ \cdot & \delta^\dagger & \cdot & -\beta^\dagger & \cdot \\ * & \cdot & * & \cdot & * \\ \cdot & -\gamma^\dagger & \cdot & \alpha^\dagger & \cdot \\ * & \cdot & * & \cdot & * \end{bmatrix} \right) \right\}$$

Crucially, the positions of the zeros in the matrices of the $\mathcal{M}(U_*^i)$ shift as i changes. Recalling that $\mathcal{M}(U_*) = \bigcap_i \mathcal{M}(U_*^i)$ we discover that

$$\mathcal{M}\left(\begin{bmatrix} \cdot & a_1 & & \\ & \ddots & \ddots & \\ & & \cdot & a_\ell \\ -a_1^\dagger & \cdot & & \\ & \ddots & \ddots & \\ & & -a_\ell^\dagger & \cdot \end{bmatrix}\right) \cong \left\{ \begin{bmatrix} \alpha_0 & & & \\ \beta_1 & \alpha_1 & & \\ & \ddots & & \\ & & \alpha_{\ell-1} & \beta_2 \\ & & & \alpha_\ell \end{bmatrix} \right\}.$$

Here, α_i is an $(e_i \times e_i)$-matrix where $e_i = \dim U_1^i$, β_1 is a $(e_1 \times e_2)$-matrix, and β_2 is a $(e_{\ell-1} \times e_\ell)$-matrix. Hence,

$$\dim \mathcal{M}(U_*) \leqslant e_1 \cdot e_2 + e_{\ell-1} \cdot e_\ell + \sum_{i=1}^{\ell} e_i^2. \tag{5.4}$$

We remark that the specific entries α_i are further constrained, since they are adjoints of the individual bimaps U_*^i. Finally, we observe that $e_i = \dim V_i - \dim V_{i-1}$, where $0 = V_0 < V_1 < \cdots < V_\ell = V$ is the fixed point flag for $\mathfrak{L}_*$, from which the desired bound $|\mathcal{M}(U_*)^\times| \leqslant |\mathfrak{L}_*|^{1/\varepsilon}$ in (5.2) follows. The bound on $\operatorname{Aut}(U_0)$ is straight-forward, since

$$\dim U_0 = \dim \mathfrak{L}_{1+s} = \sum_{i=1}^{s} d_i d_{i+s+1}.$$

Thus, $|\operatorname{Aut}(U_*)| \leqslant |\mathfrak{L}_*|^{1/\varepsilon} \cdot |\mathfrak{L}_*|^{m^2/\varepsilon}$, so Theorem 1.2 now follows from Theorem 1.1.

5.4. **Number of isomorphism types.** An immediate bound to the number of isomorphism types in $\mathcal{F}_q(d_1, \ldots, d_\ell)$ is provided by the number of partitions, which grows exponentially in the dimension. Having fixed the partition, the number of isomorphism types among dense subalgebras is also large. To see this, observe that blocks yield numerous characteristic subalgebras. We can, for example, remove

rows and columns resulting in characteristic quotients. If $\mathfrak{P}(\mathcal{F})$ is symmetric along the anti-diagonal, then we must remove both rows and columns; otherwise we can remove these independently. By removing sufficiently many rows/columns, we may assume that the partition has three parts. The number of isomorphism types for algebras based on such partitions may be estimated using arguments similar to those of Higman and Sims [2, Chapter 2] and suffice for the bound stated in Theorem 1.2.

6. Pseudo-isometries

The occasional presence of symmetry provides an opportunity to improve the complexity of our isomorphism test for graded algebras. For example, if A_* is a commutative algebra generated in degrees T, then the map $A_T \times A_T \rightarrowtail A_{T+T}$ is symmetric; if A is a Lie algebra, then $A_T \times A_T \rightarrowtail A_{T+T}$ is alternating. In such cases, the group of graded automorphisms of A_* embeds in $\Psi\mathrm{Isom}(A_T \times A_T \rightarrowtail A_{T+T})$, the group of pseudo-isometries defined in Section 2.4; typically this is a proper subgroup of $\mathrm{Aut}(A_T \times A_T \rightarrowtail A_{T+T})$. To take advantage of this observation, we must lift elements of $\mathrm{Aut}(A_{T+T})$ directly to $\Psi\mathrm{Isom}(A_T \times A_T \rightarrowtail A_{T+T})$ rather than to $\mathrm{Aut}(A_T \times A_T \rightarrowtail A_{T+T})$. Ivanyos and Qiao [14] devised an algorithm to do this.

Our objective is to solve the following problem.

PseudoIsometryCoset

Given: weakly Hermitian bimaps U_* and V_*.
Return: the coset of all pseudo-isometries of $f_*\colon U_* \to V_*$.

Once again, we first consider the restricted version of the problem that arises by insisting that maps induce the identity on the codomain. The analogue of a principal isotopism in this weakly Hermitian context is an *isometry*.

Definition 6.1. *Let U_* and V_* be weakly Hermitian bimaps. If $h\colon U_0 \to V_0$ is a linear isomorphism, then U_* and V_* are* h-isometric *if there is a pseudo-isometry $f_*\colon U_* \to V_*$ with $f_0 = h$. If $U_* = V_*$ and $h = 1$, then* $\mathrm{Isom}(U_*)$ *is the* isometry group *of U_*.*

Consider the following problem and its coset analogue.

IsometryGroup

Given: a weakly Hermitian bimap U_*.
Return: generators for $\mathrm{Isom}(U_*)$.

IsometryCoset

Given: weakly Hermitian bimaps U_*, V_* and a linear isomorphism $h\colon U_0 \to V_0$.
Return: the coset of all isometries from $U_* \to V_*$ extending h.

Polynomial-time solutions to IsometryGroup and IsometryCoset when K has odd characteristic appear in [8, Theorem 1.2] and [14], respectively. We combine these and an adaption of Algorithm 3 to solve PseudoIsometryCoset.

7. Heuristics

The family of graded Lie algebras described in Section 5 illustrates decisively the importance of the overarching "layer selection" philosophy of our approach. In this section, we assume that we have chosen a bimap $U_* = (U_2, U_1, U_0, \circ)$, and consider the problem of constructing its group of autotopisms.

In Line 4 of Algorithm 3, we have a bimap $U_* = (U_2, U_1, U_0, \circ)$ and we must choose a group G with $\mathrm{Aut}(U_*)|_{U_0} \leqslant G \leqslant \mathrm{Aut}(U_0)$. Without further information, we are forced to choose $G = \mathrm{Aut}(U_0)$, a group which is often too large to search exhaustively. In this section we introduce heuristics to cut down the order of G.

One idea is to compose $U_2 \times U_1 \rightarrowtail U_0$ with projections $\pi\colon U_0 \to K^c$ and record isotopism invariants of the resulting bimap $U_*^\pi\colon U_2 \times U_1 \rightarrowtail K^c$. Thus, we label subspaces of the dual space $U_0^\dagger = \mathrm{Hom}_K(U_0, K)$. There are $O(|K|^{c(\dim U_0)})$ subspaces of dimension c, by contrast to the $O(|K|^{(\dim U_0)^2})$ elements of $\mathrm{Aut}(U_0)$ we would otherwise be forced to list. It is therefore reasonable to list and label the former for small values of c. One then chooses G as the subgroup of $\mathrm{Aut}(U_0)$ that preserves labels.

In practice, we treat the projective geometry of $U_0^\dagger$ as a complete, colored graph, where vertices and edges are colored according to isotopism invariants. We then construct G as the automorphism group of the colored graph. Since no polynomial-time algorithm is known to construct the automorphism group of a graph, we cannot satisfactorily bound the complexity of this task. In practice, the construction of the graph is the more expensive task since the NAUTY algorithm [15] is extremely fast on generic graphs. Of course, there is no guarantee that G is a *proper* subgroup of $\mathrm{Aut}(U_0)$. We discuss this matter further in Section 8.

We now describe our labels for the points (vertices) and lines (edges) of $U_0^\dagger$. We also examine another heuristic technique called *fingerprinting* that was first introduced in [13].

7.1. **Vertex labels.** For each epimorphism $\pi\colon U_0 \to K$, we define a K-bilinear form U_*^π. In particular, if we fix bases for $U_2 \cong K^{a\times 1}$ and $U_1 \cong K^{1\times b}$, then $U_*^\pi\colon K^a \times K^b \rightarrowtail K$ is represented by its Gram matrix $D \in K^{a\times b}$ defined by

$$(e_i * e_j)\pi = e_i D e_j.$$

Base changes to U_2 and U_1 leave the rank of the Gram matrix unchanged. We use the rank of D to label $\langle\pi\rangle \in \mathbb{P}(U_0^\dagger)$; this can be computed using [21].

7.1.1. *Examples.* To illustrate subtleties arising from vertex labels, we consider two alternating bimaps $K^4 \times K^4 \rightarrowtail K^3$ specified by systems of alternating forms. Let

$$D = \left[\begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}\right],$$

$$E = \left[\begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}\right].$$

In each case the dual space $\mathrm{Hom}(K^3, K)$ determines a projective plane. For each point $P = (x_1 : x_2 : x_3)$ in the plane we obtain matrices

$$M_D(P) \equiv x_1 D[1] + x_2 D[2] + x_3 D[3] \equiv \begin{bmatrix} 0 & x_1 & 0 & 0 \\ -x_1 & 0 & x_2 & x_3 \\ 0 & -x_2 & 0 & x_1 \\ 0 & -x_3 & -x_1 & 0 \end{bmatrix} \pmod{K^\times},$$

$$M_E(P) \equiv x_1 E[1] + x_2 E[2] + x_3 E[3] \equiv \begin{bmatrix} 0 & x_1 & 0 & x_3 \\ -x_1 & 0 & x_2 & 0 \\ 0 & -x_2 & 0 & x_1 \\ -x_3 & 0 & -x_1 & 0 \end{bmatrix} \pmod{K^\times}.$$

As indicated, these matrices are unique up to nonzero scalars, and scalars do not affect their ranks. Since $M_D(P)$ and $M_E(P)$ are nonzero and alternating, the only possible ranks are 2 and 4, distinguished by whether the determinants

$$\det(M_D(P)) = x_1^4, \qquad \det(M_E(P)) = x_1^4 + 2x_1^2 x_2 x_3 + x_2^2 x_3^2$$

are zero or nonzero. Hence, the points of rank 2 for D lie on the hyperplane at infinity, namely $(0:0:1) + (0:1:0)$, while all other points have rank 4. For E, the only rank 2 points with $x_1 = 0$ satisfy $x_2 x_3 = 0$, so there are just two, namely $(0:1:0)$ and $(0:0:1)$. The remaining rank 2 points have the form $(1 : a : -1/a)$ for $a \in K^\times$. Hence, there are also $|K| + 1$ total points of rank 2, but they clearly do not lie on a line. We illustrate the situation for the field of order 3 in Figure 2.

It follows that the alternating bimaps defined by the systems D and E are not pseudo-isometric, a fact that will be transparent when we consider line labels.

7.2. **Line labels.** For each epimorphism $\lambda\colon U_0 \to K^2$, we consider the composition $U_*^\lambda\colon U_2 \times U_1 \rightarrowtail U_0 \to K^2$. Choices of bases for U_2, U_1, U_0 determine pairs of matrices associated to U_*^λ, and the label of $\langle\lambda\rangle$ is a polynomial invariant derived from this pair.

Characterizations of *indecomposable* pairs of matrices defined over the complex numbers were given by Kronecker and later for arbitrary fields by Dieudonné [10]. Using [10], Scharlau [17] characterized indecomposable skew-symmetric pairs. The reduction to indecomposable pairs is accomplished by computing a fully-refined orthogonal decomposition of U_*^λ using an algorithm of Wilson [20]. Based on Scharlau's characterization, the aforementioned polynomial invariant for alternating pairs was first introduced by Vishnivitskiĭ [19] for prime fields. It was later shown in [5, Theorem 3.22] to determine alternating pairs up to pseudo-isometry over all finite fields. The invariant may be adapted to determine all U_*^λ up to autotopism.

There are two consequences. First, the best possible labels for the lines of our projective geometry can be computed efficiently using the algorithm of [5]. Secondly, there is a far greater variety of possible line labels than there are point labels, and so the study of lines may reveal significant global constraints.

7.2.1. *Examples, revisited.* Although the bimaps D and E in Section 7.1.1 are too small to illustrate great variability, we already begin to see differences in behaviour. We illustrate this for the field of order 3 in Figure 2. For bimap D there are only two line labels. The first (colored green) labels the single line at infinity

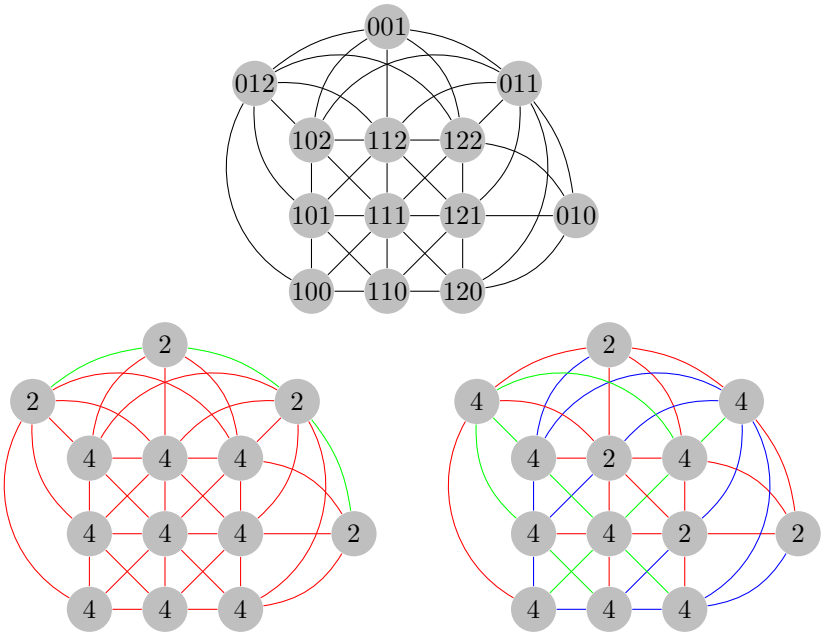

FIGURE 2. On top is the geometry $PG(2,3)$ labeled by homogeneous coordinates. The labeled and colored versions corresponding to bimaps D and E are shown on the left and right, respectively.

$(0:0:1)+(0:1:0)$, and the other lines are all labeled the same (colored red). For the bimap E there are three labels (colored blue, green, and red).

7.3. **Fingerprinting.** This process was introduced in [13] in the context of constructing automorphism groups of p-groups. Here, we examine it in more detail and generalize it to our setting.

First, consider a linear map $f\colon V \to W$. Every subspace of V is mapped to a subspace of W, and the preimage of a subspace of W is a subspace of V. This correspondence is order-preserving and allows us to compare the projective geometries of V and W.

Next, consider a bilinear map $U_*\colon U_2 \times U_1 \rightarrowtail U_0$. We compare subspaces of $U_2 \otimes U_1$ and U_0. Since most of the elements of $U_2 \otimes U_1$ are not pure tensors, the preimages that are pure tensors provide a largely *ad hoc* distribution of subspaces. This presents an opportunity to discover properties of U_* that can be used to distinguish between bimaps.

Definition 7.1. *The* left idealizer *of* $S \leqslant U_0$ *is* $\lambda(S) = \{u_2 \in U_2 \colon u_2 \circ U_1 \leqslant S\}$ *and the* right idealizer *is* $\rho(S) = \{u_1 \in U_1 \colon U_2 \circ u_1 \leqslant S\}$. *These are subspaces of* U_2 *and* U_1, *respectively.*

We collect some basic properties of idealizers (stated just for left idealizers).

Lemma 7.2. *For subspaces* S *and* T *of* U_0, *the following hold:*

(i) $S \leqslant T \implies \lambda(S) \leqslant \lambda(T)$;
(ii) $\lambda(S \cap T) = \lambda(S) \cap \lambda(T)$; *and*

(iii) $\lambda(S) + \lambda(T) \leqslant \lambda(S+T)$.

Since the subspaces $\lambda(S) + \lambda(T)$ and $\lambda(S+T)$ are typically distinct, there is an opportunity to discover hidden invariants of a bimap. Using λ and ρ, uniform incidences in the projective geometry $\mathrm{PG}(U_0)$ may be lifted to create generically irregular structures known as *subspace arrangements*. These are more general configurations than the better known *hyperplane arrangements* studied by Björner [1] and others. This can help to break symmetries, allowing us to see differences in bimaps that have no other obvious distinctions.

Let us define more carefully the objects we use. An *(affine) subspace arrangement* is a set of subspaces of a vector space. We define the *idealizer arrangements* of $U_* = (U_2, U_1, U_0, \circ)$ as follows:

$$\mathcal{A}_2 = \{\lambda(P) \colon P \in \mathrm{PG}(U_0)\} \qquad \mathcal{A}_1 = \{\rho(P) \colon P \in \mathrm{PG}(U_0)\}.$$

We can build other arrangements for U_* by shuffling coordinates. One way to discover $\mathrm{Aut}(U_*)$-invariant substructures is to compute the intersection numbers of the arrangement up to some rank. This is essentially what is described as a *fingerprint* in [13] and the algorithm described there is polynomial in $|U_0|$. One could, however, use more refined tools such as the Möbius function $\mu \colon \mathrm{PG}(U_0) \times \mathrm{PG}(U_0) \to \mathbb{Z}$, where

$$\mu(P,Q) = \begin{cases} 1 & \lambda(P) = \lambda(Q), \\ -\sum_{\lambda(P) \leqslant X < \lambda(Q)} \mu(\lambda(P), X) & \lambda(P) < \lambda(Q), \\ 0 & \text{otherwise.} \end{cases}$$

8. Implementation and performance

We have implemented our algorithms in Magma. The implementation utilizes various packages—all publicly available on GitHub—that have been developed by the authors and their collaborators [24].

The examples in Sections 7.1.1 and 7.2.1 show how our local heuristics expose global structures. Here we briefly report on experiments with our implementation that demonstrate their impact.

8.1. **Twisted Heisenberg groups.** Fix a prime p and odd integer $k > 1$. Put $q := p^k$, $A := \mathbb{F}_q$, and choose $\sigma \in \mathrm{Gal}(\mathbb{F}_q)$. Define a product $\cdot$ on A, where $x \cdot y = xy + ix^\sigma y^{\sigma^2}$ for $x, y \in A$. This turns A into a *twisted Albert algebra*, a nonassociative finite division algebra. Finally, define a bimap $\bullet \colon A^2 \times A^2 \rightarrowtail A$ by

$$u \bullet v = u \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} v^\dagger.$$

Over the ground field $\mathbb{F}_p$, this gives a bimap $\mathbb{F}_p^{2k} \times \mathbb{F}_p^{2k} \rightarrowtail \mathbb{F}_p^k$, which can be encoded for computation as a system of k alternating forms of degree $2k$. These bimaps arise from the *twisted Heisenberg groups*

$$H(A) = \left\{ \begin{bmatrix} 1 & \alpha & \gamma \\ 0 & 1 & \beta \\ 0 & 0 & 1 \end{bmatrix} : \alpha, \beta, \gamma \in A \right\}.$$

For various choices $1 < e \leqslant k$, we now compose $\bullet$ with random $\varphi \in \mathrm{Hom}(\mathbb{F}_{p^k}, \mathbb{F}_{p^e})$ to obtain bimaps $\circ \colon \mathbb{F}_p^{2k} \times \mathbb{F}_p^{2k} \rightarrowtail \mathbb{F}_p^e$, where $u \circ v := u \bullet^\varphi v = (u \bullet v)\varphi$. The objective is to construct the group $\Gamma\mathrm{Isom}(\circ)$. The default is to use the algorithm outlined in

Section 6 which requires us to list $\mathrm{GL}(e,p)$. An alternative is to build the projective geometry $\mathrm{PG}(e,p)$ and label its points and lines, as described in Section 7.

Table 1 records information from experiments with our implementation. For different choices of p, k and e, it records: the numbers of points and lines in $\mathrm{PG}(e,p)$; the approximate order of $\mathrm{GL}(e,p)$; the order of the subgroup Ω of $\mathrm{GL}(e,p)$ preserving the point and line labels; and finally the order of the subgroup Ψ of Ω that lifts to pseudo-isometries of $\circ$. For each choice of parameters, we performed 10 random trials and recorded the $|\Omega|$ and $|\Psi|$ that occurred most often. (In fact, these numbers were the same for all runs.)

p	k	e	# points	# lines	$\lvert\mathrm{GL}(e,p)\rvert$	$\lvert\Omega\rvert$	$\lvert\Psi\rvert$
3	5	3	13	13	$\approx 10^4$	16	2
3	5	4	40	130	$\approx 10^8$	10	10
3	6	4	40	130	$\approx 10^8$	1	1
5	5	3	31	31	$\approx 10^7$	1	1
5	5	4	156	806	$\approx 10^{11}$	20	20

TABLE 1. Experiments for the twisted Heisenberg groups

We remark that other weaker (but more efficiently computed) invariants can be used to label the lines in our projective geometries. Often these suffice to discover structures that must be preserved by pseudo-isometries (or by autotopisms in more general applications). However, the weaker invariants do not distinguish quotients of twisted Heisenberg groups: all resulting line labels are identical. But, by using the invariants described in Section 7.2, the problem breaks completely.

8.2. **Random nilpotent Lie algebras of class 2.** If $U_* = (U_2, U_1, U_0, \circ)$ is a generic bimap—which we loosely define to be one specified by an arbitrary selection of structure constants—then the group induced by $\mathrm{Aut}(U_*)$ on U_0 is usually very small, and is often trivial. In such cases, local invariants (even the weaker ones) almost always cut down to an overgroup containing the correct group as a subgroup of very small index.

In Table 2, we mimic the experiments and reporting of Section 8.1, but this time we generate alternating bimaps $\circ\colon \mathbb{F}_p^d \times \mathbb{F}_p^d \rightarrowtail \mathbb{F}_p^e$ by selecting e random skew-symmetric $d \times d$ matrices over $\mathbb{F}_p$. Once again, we select the most commonly occurring $|\Omega|$ and $|\Psi|$ from 10 runs with each choice of parameters (d, p, e).

d	p	e	# points	# lines	$\lvert\mathrm{GL}(e,p)\rvert$	$\lvert\Omega\rvert$	$\lvert\Psi\rvert$
10	3	3	13	13	$\approx 10^4$	1	1
20	3	3	13	13	$\approx 10^4$	1	1
20	3	4	40	130	$\approx 10^8$	1	1
10	3	5	121	1210	$\approx 10^{11}$	1	1
20	5	3	31	31	$\approx 10^6$	1	1
10	5	4	156	806	$\approx 10^{11}$	1	1
10	5	5	781	20306	$\approx 10^{17}$	1	1

TABLE 2. Experiments for Lie algebras of class 2

These tests suggest that, for generic bimaps, $\Psi\mathrm{Isom}(\circ)$ acts trivially on its codomain. The variation of the dimension, d, of the domain space has little impact on the outcome, but increasing e introduces more constraints and therefore makes it increasingly likely that $\Psi\mathrm{Isom}(\circ)$ acts trivially. As we see, the local invariants usually detect when this is the case.

8.3. **Do the heuristics always work?** The striking practical success of the local invariants raises the obvious question of whether we can strengthen the theoretical complexity of Theorem 1.1. If we label just points and lines, the answer is no: there exist alternating bimaps U_* for which all points and lines are labeled identically, yet the group induced by $\Psi\mathrm{Isom}(U_*)$ on its codomain U_0 is a proper subgroup of $\mathrm{Aut}(U_0)$; one such example appears in [18].

References

[1] Anders Björner, *Subspace arrangements*, First European Congress of Mathematics, Vol. I (Paris, 1992), Progr. Math., vol. 119, Birkhäuser, Basel, 1994, pp. 321–370. MR1341828

[2] Simon R. Blackburn, Peter M. Neumann, and Geetha Venkataraman, *Enumeration of finite groups*, Cambridge Tracts in Mathematics, vol. 173, Cambridge University Press, Cambridge, 2007. MR2382539

[3] W. Bosma, J. Cannon, and C. Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput. **24** (1997), no. 3-4, 235–265. MR1484478

[4] Peter A. Brooksbank and Eugene M. Luks, *Testing isomorphism of modules*, J. Algebra **320** (2008), no. 11, 4020–4029. MR2464805 (2009h:16001)

[5] Peter A. Brooksbank, James B. Wilson, and Joshua Maglione, *A fast isomorphism test for groups whose Lie algebra has genus* 2, J. Algebra **473** (2017), 545–590. MR3591162

[6] Peter A. Brooksbank and E. A. O'Brien, *Constructing the group preserving a system of forms*, Internat. J. Algebra Comput. **18** (2008), no. 2, 227–241. MR2403820 (2009g:20020)

[7] Peter A. Brooksbank and James B. Wilson, *Groups acting on tensor products*, J. Pure Appl. Algebra **218** (2014), no. 3, 405–416. MR3124207

[8] ———, *Computing isometry groups of Hermitian maps*, Trans. Amer. Math. Soc. **364** (2012), no. 4, 1975–1996. MR2869196

[9] David G. Cantor and Hans Zassenhaus, *A new algorithm for factoring polynomials over finite fields*, Math. Comp. **36** (1981), no. 154, 587–592.

[10] Jean Dieudonné, *Sur la réduction canonique des couples de matrices*, Bull. Soc. Math. France **74** (1946), 130–146. MR0022826 (9,264f)

[11] Bettina Eick, *Computing the automorphism group of a solvable Lie algebra*, Linear Algebra Appl. **382** (2004), 195–209. MR2050106

[12] ———, *Computing automorphism groups and testing isomorphisms for modular group algebras*, J. Algebra **320** (2008), no. 11, 3895–3910.

[13] Bettina Eick, C. R. Leedham-Green, and E. A. O'Brien, *Constructing automorphism groups of p-groups*, Comm. Algebra **30** (2002), no. 5, 2271–2295. MR1904637 (2003d:20027)

[14] Gábor Ivanyos and Youming Qiao, *Algorithms based on $*$-algebras, and their applications to isomorphism of polynomials with one secret, group isomorphism, and polynomial identity testing*, SIAM Journal on Computing.

[15] Brendan D. McKay, *Practical Graph Isomorphism*, Congr. Numer. **30** (1981), 45–87.

[16] Joshua Maglione, *Efficient characteristic refinements for finite groups*, J. Symbolic Comput. **80** (2017), 511–520. MR3574524

[17] Rudolf Scharlau, *Paare alternierender Formen*, Math. Z. **147** (1976), no. 1, 13–19. MR0419484 (54 #7505)

[18] Libero Verardi, *Semi-extraspecial groups of exponent p*, Ann. Mat. Pura Appl. **148** (1987), no. 4, 131–171. MR0932762 (89h:20033)

[19] A. L. Vishnevetskiĭ, *A system of invariants of certain groups of class* 2 *with commutator subgroup of rank two*, Ukrain. Mat. Zh. **37** (1985), no. 3, 294–300, 403. MR795568 (86k:20033)

[20] James B. Wilson, *Finding central decompositions of p-groups*, J. Group Theory **12** (2009), no. 6, 813–830. MR2582050

[21] ———, *Optimal algorithms of Gram-Schmidt type*, Linear Algebra Appl. **438** (2013), no. 12, 4573–4583. MR3039211
[22] ———, *Division, adjoints, and dualities of bilinear maps*, Comm. Algebra **41** (2013), no. 11, 3989–4008. MR3169502
[23] ———, *On automorphisms of groups, rings, and algebras*, Comm. Algebra **45** (2017), no. 4, 1452–1478. MR3576669
[24] Peter A. Brooksbank, Joshua Maglione, E.A. O'Brien, and James B. Wilson, *TheTensor.Space: Repositories for multilinear algebra and isomorphism tests*, `https://github.com/thetensor-space`, 2019.

Brooksbank, Department of Mathematics, Bucknell University, Lewisburg, PA 17837, USA

O'Brien, Department of Mathematics, University of Auckland, Private Bag 92019, Auckland, New Zealand

Wilson, Department of Mathematics, Colorado State University, Fort Collins, CO 80523, USA