hexagon:2611.01261v2

Riemannian Symmetric Graphs

Structure, Classification and Computational Complexity

Sirawit Pongnakintr
ETH Zürich*

Version 2, 29 November 2026

Abstract

We study Riemannian symmetric graphs, which are graphs admitting local involutions at vertices and edges in such a way that they obey a common Loos conjugation law. We characterize compatibility by central involutions in the stabilizers of a supporting group and prove that equivariance survives four-point closure. Recognizing either local property independently is NP-complete. We reduce 3-CNF satisfiability to full compatibility in deterministic time $2^{O((s+1)^{1/4})}(|w|+1)^{O(1)}$, where w is the encoded formula and s counts its variables plus clauses. The reduction produces a simple graph of order n together with a binary chart containing its full automorphism group and generators for that group. Recognition in such a chart takes $2^{O((\log n)^4)}$ time. For primitive witness actions we determine the three possible forms of the vertex operation and give the edge-orbit criterion that recovers the full system from that operation. We realize the odd affine and simple group core forms and give explicit involution-class realizations.

AI Disclosure and Methodology: This project expanded from an independent human curiosity, originally discussed only with ChatGPT 5.5, ChatGPT 5.6 Sol, ChatGPT 5.6 Astra, Claude Sonnet 5, Claude Opus 5, and Claude Opus 5.5 in the first draft. The first draft proved that VIG recognition is NP-complete and provided a flawed attempt at EIG recognition hardness. As AI systems become more capable, we performed a substantial AI-powered revision which fixed the error and provided a further extensive exploration. In this revision, the main AI systems are ChatGPT GPT-6 Chat mode, ChatGPT GPT-6.2 Chat mode, and Claude Opus 5.5 for high-level discussion. The main exploration loop is performed by two independent OpenAI ChatGPT 6.2 Sol (high) Codex agents, directed by the human author. Codex assisted with developing the proof arguments, computational checks, literature comparison, and preparing the manuscript. Separate ChatGPT GPT-6.2 review identified places requiring further verification. The local EIG and fourth-root proofs warrant independent end-to-end review beyond the finite checks reported here. The human author does not claim full understanding of the results presented here.

Acknowledgements. The human author thanks Jittat Fakcharoenphol for his inspiring advice and previous supervision which brought the human author into the graph isomorphism problem literature. The human author also thanks Alessandra Iozzi for her lectures on Symmetric Spaces. Computational resources are significantly supported by the ETH Euler cluster.

Contents

1 Introduction **2**
1.1 Contributions 3
1.2 Open Problems 3

*MSc Student, Department of Mathematics

2 Definitions and basic consequences **3**

3 Supporting groups, equivariance and closure **6**
3.1 Four-point closure of a supporting group . . . 8
3.2 An obstruction to compatibility . . . 8

4 Complexity of the local recognition problems **9**

5 Global compatibility and binary charts **9**
5.1 Recognition in a supplied chart . . . 10

6 Primitive systems **12**
6.1 The edge-orbit criterion . . . 13
6.2 Explicit realizations . . . 14

A Proofs of local hardness **15**
A.1 A bounded-degree source for the vertex reduction . . . 15
A.2 Vertex-involutive graph recognition . . . 16
A.3 Edge-involutive graph recognition . . . 17
A.3.1 A translation-invariant syndrome . . . 17
A.3.2 A uniform computation on the corner positions . . . 18
A.3.3 The group on two copies of the positions . . . 19
A.3.4 Ordered-pair closure and the pinned exchange . . . 21
A.3.5 From a closed transporter to a graph . . . 22
A.3.6 Bipartite instances and completion of the reduction . . . 24

B Proof of the fourth-root reduction **25**
B.1 Compatibility-preserving color removal . . . 25
B.2 Scalar derivative constraints . . . 26
B.3 The reduction . . . 26

C Finite verification and reproducibility **33**
C.1 EIG syndrome and orientation checks . . . 33
C.2 Fourth-root algebraic checks . . . 35

1 Introduction

In the Riemannian globally symmetric spaces literature, we study Riemannian manifolds which admit geodesic symmetries s_p at each point p which reverses geodesics, locally fixes only p, and can be extended globally.

Historically, Cartan has successfully performed a classification program for Riemannian globally symmetric spaces through algebraic reformulations using Riemannian symmetric pair, orthogonal symmetric Lie algebra decomposition, and root systems. Cartan's classification of Riemannian symmetric spaces and Loos's algebraic treatment of their point symmetries motivate a finite model in which conjugating a reflection transports its center [Car26, Car27, Loo67, Loo69]. The same law appears in Takasaki's kei and in the theory of finite symmetric sets [Tak43, KNN76]. Connected quandles admit a group description in terms of central elements of point stabilizers, and related graph constructions arise in the study of generalized Cayley graphs [HSV16, Ta26]. In the model studied here, an edge can also be a center. Its assigned involution reverses the edge, and the vertex and edge assignments must be transported together.

1.1 Contributions

Our specific contributions are:

Identifying the VIG, EIG, and RSG graph classes as discrete graph-theoretic analogues of symmetric spaces. We give the definitions and basic consequences in Section 2. We identify the discrete analogue and reformulate the notion of *kei* or *involutive quandle* in a graph-theoretic form to study them.

Hardness of VIG/EIG recognition, and partial results on the hardness of RSG recognition. VIG and EIG recognition are NP-complete (Section 4). Section 5 gives a fourth-root reduction and an algorithm for recognition in a supplied chart. By Corollary 5.4, RSG recognition in time $2^{O((\log n)^{4-\varepsilon})}n^{O(1)}$ for fixed $0 < \varepsilon < 4$ on the specified triples of graph order n would yield deterministic time $2^{O((s+1)^{\max\{1/4,1-\varepsilon/4\}})}(|w|+1)^{O(1)}$ for 3-SAT, where w is the encoded formula and s counts its variables plus clauses. Polynomial-time RSG recognition would give the exponent $1/4$ in the 3-SAT bound. Either result would refute the Satisfiability Hypothesis (SH) [GBD+26] on formulas with consecutively numbered variables.

Primitive classification of RSG. Inspired by Cartan's classification program, we establish a classification of primitive RSG in Section 6. Full classification remains open.

1.2 Open Problems

Our main curiosity that still hasn't yet been closed after extensive AI assistance is the full knowledge about the hardness on RSG recognition. In particular, we are curious whether RSG recognition is NP-hard, or if it is solvable in quasipolynomial time, or neither.

The RSG recognition problem illustrates an interesting tension between computational expressivity and algebraic rigidity. Although local involutive symmetry can encode NP-complete problems, global compatibility imposes additional algebraic constraints on the possible witnesses. It is not clear whether these constraints lead to substantially faster recognition algorithms or whether they retain enough expressive power to encode arbitrary NP-complete problems through polynomial-time reductions. We believe that understanding this tension is an interesting research problem in its own right.

Another curiosity on the mathematical side is a full discrete analogue of Cartan's program on the classification of Riemannian globally symmetric spaces. We believe such a classification has its own mathematical merits regardless of complexity results.

2 Definitions and basic consequences

All graphs are finite, simple and undirected unless stated otherwise. Write $V = V(G)$ and $E = E(G)$. We will use the symbol $\mathcal{I}$ to denote an index set, which will be either V, E, or $V \sqcup E$. An involution has order exactly two. A graph automorphism acts on unordered edge indices by its action on their endpoints.

Definition 2.1 (Vertex Involutive Graph)**.** Let $G = (V, E)$ be a graph. A *vertex witness* is an involutive[1] automorphism $s_u \in \mathrm{Aut}(G)$ such that $\mathrm{Fix}(s_u) \cap N[u] = \{u\}$, where $N(u) := \{v \in V : uv \in E\}$ and $N[u] := N(u) \cup \{u\}$. A graph G is a vertex involutive graph (VIG) if it admits vertex witnesses at all vertices. We write $\mathcal{VIG}$ for the class of all VIGs.

Proposition 2.2. *Let G be a VIG. Then for all $u \in V(G)$, $\deg_G(u)$ is even. In particular, each connected component is Eulerian, and hence the graph is bridgeless.*

[1]Here, s being involutive means $s \neq \mathrm{id}$ and $s \circ s = \mathrm{id}$.

Proof. Let $u \in V(G)$. We claim that $s_u(N(u)) = N(u)$. Indeed, for all $v \in N(u)$, the edge uv is in $E(G)$. Since s_u is an automorphism, the edge $s_u(u)s_u(v)$ is also in $E(G)$. But $s_u(u) = u$ so $us_u(v) \in E(G)$, so $s_u(v) \in N(u)$. This shows that $s_u(N(u)) \subseteq N(u)$. By bijectivity of s_u, the sets coincide, so $s_u(N(u)) = N(u)$. This means $s_u|_{N(u)}$ is a permutation.

Now, the fact that $\mathrm{Fix}(s_u) \cap N[u] = \{u\}$ means $\mathrm{Fix}(s_u) \cap N(u) = \varnothing$, so $s_u|_{N(u)}$ is a permutation with no fixed point, and $s_u \circ s_u = \mathrm{id}$ means all cycle have length 2. This shows that $\deg u = |N(u)| = 2 \cdot \#$ of cycles is even. □

The following result shows an import of the intuition from the Riemannian symmetric space literature: a *transvection* "transports" a cut to another.

Proposition 2.3. *Let G be a VIG. Then each connected component of G is either a singleton or 2-connected.*

This uses the following simple lemma.

Lemma 2.4. *Let G be a VIG. Let $(s_v)_{v \in V(G)}$ be the collection of involutive automorphisms satisfying the definition of VIG. The following statement works for any choice of such $(s_v)_{v \in V(G)}$: If u is a cut vertex of G and $v \in N(u)$, then $s_v(u)$ is also a cut vertex of G.*

Proof. Let $w := s_v(u)$. Since u is a cut vertex of G, let A be the connected component of u in G, viewed as a subgraph, then the graph $A \setminus \{u\}$ becomes disconnected. In particular, there exists some $t \in V(A) \setminus \{u, v\}$ such that t is not reachable from v in $G \setminus \{u\}$ but t is adjacent to u.

Now since $tu \in E(G)$ and $s_v \in \mathrm{Aut}(G)$, we see that $s_v(t)s_v(u) \in E(G)$ as well. Let $x := s_v(t)$ to see that $wx \in E(G)$.

Suppose now (for the sake of contradiction) that w is not a cut vertex then since $vw, wx \in E(G)$, we see that there is a simple path joining v to x in $G \setminus \{w\}$, say, $v = p_0 \to p_1 \to p_2 \to \cdots \to p_k = x$ with $p_i \neq w$ for all i. Since s_v is an automorphism, we see that

$$s_v(p_0) \to s_v(p_1) \to \cdots \to s_v(p_k)$$

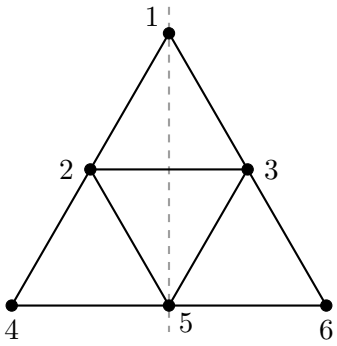
is also a path of G from $s_v(p_0) = s_v(v) = v$ to $s_v(p_k) = s_v(x) = t$, not containing $s_v(w) = u$. This contradicts the fact that t is not reachable from v in $G \setminus \{u\}$. This proves that w is a cut vertex. □

Now we're ready to go to 2.3.

Proof of 2.3. Without loss of generality, we can assume G to be connected and is not a singleton. Suppose it is not 2-connected. Construct the block-cut tree of the graph G. Since it is a tree, there is a leaf component containing a cut vertex. Let $B \leq G$ be the block subgraph corresponding to that leaf, and let $c \in V(B)$ denote the cut vertex. By definition, such block cannot be a single vertex (otherwise it's just the cut, not actually a block). Pick a neighboring vertex $u \in N_B(c)$, which is not a cut (so all neighbors of u in G are actually in B, in particular, $s_u(c) \in V(B)$). Then $s_u(c)$ is also a cut, by lemma 2.4. Since s_u cannot fix c, we see that $s_u(c)$ is an actual different cut than c in B. This means u is adjacent to two cuts (in the same block B), a contradiction to the assumption that B is a leaf component (containing only one cut vertex). □

One might hope that with such regularity conditions in the definition of a VIG, the graph must become highly symmetric in the sense that it is vertex-transitive or edge-transitive. However, this is not the case, considering the following example.

Example 2.5. Let $V = \{1, 2, 3, 4, 5, 6\}$ and $E = \{12, 13, 23, 24, 25, 35, 36, 45, 56\}$. This is a VIG. In the embedding below, the dashed line indicates a reflection axis for s_1 and s_5. Other involutive automorphisms can be defined similarly. One can check that this graph is neither vertex-transitive nor edge-transitive!



Now we consider another variant of such involutive symmetry.

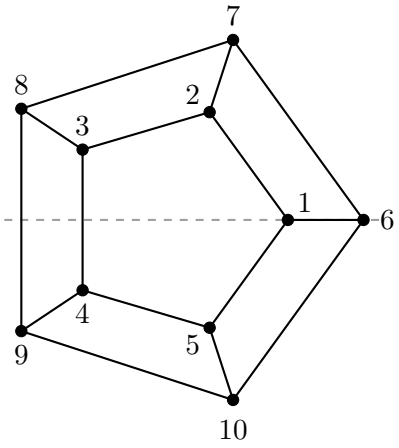
Definition 2.6 (Edge Involutive Graph)**.** Let $G = (V, E)$ be a graph. For an edge $e = uv \in E$, an *edge witness* is an involutive automorphism $s_e \in \mathrm{Aut}(G)$ such that $s_e(u) = v$ (and so $s_e(v) = u$). We say that a graph G is an edge involutive graph (EIG) if it admits edge witnesses at all edges. We write $\mathcal{EIG}$ for the class of all EIGs.

In fact, this variant is vertex-transitive.

Proposition 2.7. *Let G be a connected EIG. It is vertex-transitive.*

Proof. Let $u, v \in V(G)$ be given. We claim that there exists an automorphism $\sigma \in \mathrm{Aut}(G)$ sending u to v. Indeed, since G is connected, there exists a simple path $p_0 p_1 \dots p_k$ with $p_0 = u$ and $p_k = v$. Put $\sigma := s_{p_{k-1}p_k} \circ s_{p_{k-2}p_{k-1}} \circ \cdots \circ s_{p_0 p_1} \in \mathrm{Aut}(G)$. □

Example 2.8. Let $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and
$E = \{12, 23, 34, 45, 15, 16, 27, 38, 49, 5(10), 67, 78, 89, 9(10), 6(10)\}$. The dashed line indicate a reflection axis for s_{34}. For the edge $\{i, i+5\}$ ($i \in \{1, 2, 3, 4, 5\}$), we let $s_{\{i,i+5\}}$ swap the inside pentagon with the outside pentagon. Observe that for EIG we cannot hope for the Eulerian property!



Each connected component of an EIG is vertex-transitive. A connected vertex-transitive graph of degree k on at least two vertices has edge connectivity k [GR01, Lemma 3.3.3]. If it has at least three vertices, its vertex connectivity is at least $2(k+1)/3$ [GR01, Theorem 3.4.2].

Example 2.9. The Doyle–Holt graph [Doy76] [Hol81] is known to be vertex-transitive, edge-transitive, but not arc-transitive. If it were to be an EIG, then by edge-transitivity, flipping edges would imply arc-transitivity. Therefore, this is an example of a vertex-transitive graph which is not an EIG.

Definition 2.10. Let G be a graph. Let $\mathcal{I} := V \sqcup E$. A *full Riemannian system* is a family $(s_a)_{a\in\mathcal{I}}$ of vertex and edge witnesses satisfying

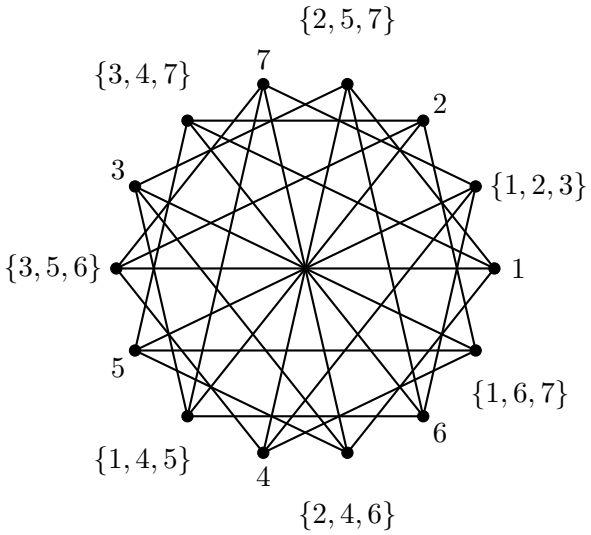
$$s_{s_a(b)} = s_a s_b s_a^{-1} \qquad (a, b \in \mathcal{I}). \tag{1}$$

A *Riemannian symmetric graph* (RSG) is a graph that admits a full Riemannian system.

A compatible vertex system has the same definition as above but replacing $\mathcal{I} := V$. A compatible edge system is the same but with $\mathcal{I} := E$.

Each index is fixed by its own witness. Thus $a \triangleright b = s_a(b)$ defines a left involutory quandle on $\mathcal{I}$, with the vertex and edge indices distinguished. A full system restricts to a vertex/edge compatible system.

Example 2.11. Let $F := \{\{1,2,3\},\{2,5,7\},\{3,4,7\},\{3,5,6\},\{1,4,5\},\{2,4,6\},\{1,6,7\}\}$ denote the Fano plane. Let $V = \{1,2,3,4,5,6,7\} \cup F$. Let $E = \{uv \colon u \in \{1,\dots,7\}, v \in F, u \notin v\}$ denote the edges. The corresponding graph $G = (V, E)$ is the graph of non-incidence of the Fano plane. One can check that this graph is both VIG and EIG but not RSG.



Proposition 2.12. *Every connected RSG except the singleton is regular of positive even degree.*

Proof. Apply Proposition 2.7 to see regularity and apply Proposition 2.2 to see even degree. □

With the order-two convention, the graph on one vertex fails the vertex condition. On disconnected graphs witnesses can act on components other than the one containing their index. Every cycle graph is an RSG.

3 Supporting groups, equivariance and closure

Definition 3.1. Let G be a graph. Let $\mathcal{I}$ be V, E, or $V \sqcup E$. Let $s_\bullet := (s_a)_{a\in\mathcal{I}}$ be a system. An *equivariant support* of $s_\bullet$ is a subgroup $H \leq \mathrm{Aut}(G)$ such that for each $a \in \mathcal{I}$, $s_a \in H$,

and under which the assignments are equivariant, i.e., the following diagram commutes for all $\sigma \in H$.

$$\begin{array}{ccc} \mathcal{I} & \xrightarrow{a \mapsto s_a} & H \\ {\scriptstyle a \mapsto \sigma(a)}\downarrow & & \downarrow{\scriptstyle h \mapsto \sigma h \sigma^{-1}} \\ \mathcal{I} & \xrightarrow{a \mapsto s_a} & H \end{array}$$

The *witness group* is the subgroup generated by the witnesses, i.e. $\langle s_a : a \in \mathcal{I} \rangle$.

Theorem 3.2 (Supporting subgroup criterion)**.** *Let the index set $\mathcal{I}$ be $V \sqcup E$, E, or V for a full, edge-only, or vertex-only system, respectively. A subgroup $H \leq \mathrm{Aut}(G)$ supports some system indexed by $\mathcal{I}$ if and only if for each H-orbit on $\mathcal{I}$, there exists a representative a of that orbit such that the stabilizer center $Z(\mathrm{Stab}_H(a))$ contains some valid witness of a. A system always has its witness group as an equivariant support. On a connected graph, every support of a full or edge-only system is transitive on the vertices.*

Proof. Given a system $s_\bullet$ and an equivariant support $H \leq \mathrm{Aut}(G)$. Let $a \in \mathcal{I}$ be arbitrary. We have $s_a \in \mathrm{Stab}_H(a)$. Equivariance implies $\sigma s_a \sigma^{-1} = s_{\sigma(a)}$, i.e. $\sigma s_a = s_{\sigma(a)}\sigma$, for all $\sigma \in H$. If we restrict to $\sigma \in \mathrm{Stab}_H(a)$, we get $\sigma(a) = a$ and so $\sigma s_a = s_a \sigma$ for all $\sigma \in \mathrm{Stab}_H(a)$, i.e. $s_a \in Z(\mathrm{Stab}_H(a))$.

Conversely, for each representative a of an H-orbit on $\mathcal{I}$, we pick a witness $s_a \in Z(\mathrm{Stab}_H(a))$. Define $\sigma_\bullet$ to be the unique system that $\sigma_{h(a)} = h s_a h^{-1}$ for each representative a and for each $h \in H$. If $h(a) = h'(a)$ then $h^{-1}h' \in \mathrm{Stab}_H(a)$ and $(h s_a h^{-1})^{-1}(h' s_a h'^{-1}) = h s_a^{-1} h^{-1} h' s_a h'^{-1}$. But by assumption, $s_a \in Z(\mathrm{Stab}_H(a))$ so $h s_a^{-1} h^{-1} h' s_a h'^{-1} = h s_a^{-1} s_a h^{-1} h' h'^{-1} = \mathrm{id}$. This shows that $h s_a h^{-1} = h' s_a h'^{-1}$, and so the definition of this system $\sigma_\bullet$ is well-defined, and that each element of $\sigma_\bullet$ belongs to H.

By the very definition of the system and that $\sigma_a = s_a$, we see that $\sigma_{h(a)} = h \sigma_a h^{-1}$ for all representative a and each $h \in H$, we see that the system is H-equivariant at each representative. Now for general $b \in \mathcal{I}$, its orbit $\mathrm{Orb}_H(b)$ includes a representative a such that $a = h(b)$ for some $h \in H$, so

$$\sigma_{h'(b)} = \sigma_{h'(h^{-1}(a))} = \sigma_{(h'h^{-1})(a)} = h'h^{-1}\sigma_a (h'h^{-1})^{-1} = h'h^{-1}\sigma_a h h'^{-1} = h'\sigma_{h^{-1}(a)}h'^{-1} = h'\sigma_b h'^{-1}$$

for all $h' \in H$. This proves the equivariance everywhere, and hence $\sigma_\bullet$ is a system that is supported by H.

The Loos identity establishes equivariance under each assigned involution. Composing these identities along words in the generators establishes equivariance under the witness group.

When edge witnesses are present on a connected graph, take a path $v_0, \ldots, v_k$. The successive edge witnesses carry v_0 to v_k, and their product belongs to H. □

This allows us to characterize the existence of compatible vertex systems, compatible edge systems, and full Riemannian systems in terms of supporting subgroups. In particular, G is an RSG if and only if there exists a subgroup $H \leq \mathrm{Aut}(G)$ satisfying the stabilizer-center criterion for $\mathcal{I} = V \sqcup E$.

Proposition 3.3. *VIG, EIG, and RSG recognition are in* NP.

Proof. A certificate lists one permutation per required index. Check its order, adjacency preservation, and the local vertex or edge condition. For RSG also check every ordered pair of indices in (1). The sizes of the lists and all the checks are polynomial in the graph order. □

3.1 Four-point closure of a supporting group

For a permutation group H on the vertices, its k-closure $H^{(k)}$ is defined as the subgroup of permutations that preserve each H-orbit (setwise) on ordered k-tuples. That is,

$$H^{(k)} := \bigcap_{u_1,\dots,u_k \in V} \mathrm{Stab}_{\mathrm{Sym}(V)}(\mathrm{Orb}_H((u_1,\dots,u_k))).$$

Proposition 3.4. *A system equivariant under H is also equivariant under its four-point closure $H^{(4)}$. This holds for edge-only and full systems. If the witnesses generate H, then H is normal in $H^{(4)}$. In particular a subgroup maximal among the supporting subgroups of a graph is four-point closed. The exponent four cannot in general be replaced by three, even for compatible edge systems.*

Proof. The closure preserves the graph because it preserves the orbits on ordered pairs. At an edge $e = \{u, v\}$, use the tuple $(u, v, x, \sigma_e(x))$ for arbitrary x. If $g \in H^{(4)}$, take $h \in H$ agreeing with g on this tuple. Then $h(e) = g(e)$, and equivariance under H gives

$$g\sigma_e(x) = h\sigma_e(x) = \sigma_{g(e)}(g(x)).$$

Thus g transports the edge witnesses correctly. For a vertex u, the tuple $(u, x, \sigma_u(x))$ gives the same argument using only three points. If H is witness-generated, these transport equations show that $H^{(4)}$ normalizes H. Since the larger group still supports the system, maximality implies that the group equals its closure.

For sharpness take $I = \mathbb{F}_2^d$, $L = \mathbb{F}_2^r$, $d \geq 3$, $r \geq 1$, and put all edges between distinct position blocks of $I \times L$. Let H consist of the permutations $(x, u) \mapsto (x + a, u + f(x))$ with $a \in I$ and $f : I \to L$ affine. At the representative edge between $(0, 0)$ and $(a, 0)$, $a \neq 0$, use the pure shift by a. Any pointwise endpoint fixer is an affine profile vanishing at $0, a$, and is invariant under translation by a. This reverser is therefore central in the edge stabilizer. All edges with the same position difference form one orbit because affine profiles can take arbitrary values at two distinct positions. The two-point orbits force preservation of all ordered position differences and all within-block label differences. They are preserved by every position translation with an arbitrary label profile, since an affine profile can take arbitrary values at two distinct positions. Thus $H^{(2)} = L^I \rtimes I$. Over $\mathbb{F}_2$ every three distinct positions are affinely independent, so arbitrary values at three positions also extend to an affine profile. Thus $H^{(3)} = H^{(2)}$.

In this larger group a reverser of an edge over direction $a \neq 0$ must shift the positions by a. A nonzero profile supported at one position outside the two endpoint blocks fixes the edge pointwise. Conjugation by any such reverser shifts its support by a, so the reverser cannot centralize that pointwise stabilizer. Hence the three-point closure admits no equivariant edge system, despite the system in H. In this example $H^{(4)} = H$: the orbit constraints on four positions in an affine plane force each profile to be affine. □

3.2 An obstruction to compatibility

Theorem 3.5. *The connected Kneser graph* $\mathrm{KG}(7, 3)$ *is EIG and VIG. It admits no compatible edge system and hence is not RSG.*

Proof. The vertices are the triples in $\{1, \dots, 7\}$, with disjoint triples adjacent. Two triples that differ in one point have a common neighbor, namely the complement of their union. Replacing one point at a time connects every pair of triples, so the graph is connected.

For two disjoint triples, pair their points bijectively and exchange the three pairs to obtain an edge involution. For a triple U, take a double transposition on its four-point complement and

fix U pointwise. Each neighbor of U is specified by the point it omits from that complement, and the double transposition moves that point. This gives a vertex involution.

By the Erdős–Ko–Rado theorem and its equality case [EKR61, HM67], the maximum independent sets are the seven families of triples through a given ground point. Every automorphism permutes these families, and a triple is determined by the three families that contain it. Therefore $\mathrm{Aut}(\mathrm{KG}(7,3)) = S_7$ in its action on triples.

Suppose a compatible edge system generates $H \leq S_7$. The edge path argument makes H transitive on the 35 triples, so 35 divides $|H|$. Cauchy's theorem gives a 7-cycle and a 5-cycle on the ground set. The 7-cycle makes H transitive on the ground set and hence primitive, as the degree is prime. Jones's theorem on a cycle fixing two points gives $A_7 \leq H$, since $7-1$ is not a prime power [Jon14, Theorem 1.2]. Every involution exchanging two disjoint triples is a product of three transpositions, so H also contains an odd permutation. Thus $H = S_7$. The setwise stabilizer of an edge formed by two disjoint triples contains a pure symmetric group on each triple. Every element reversing the edge interchanges these two subgroups and therefore fails to centralize the edge stabilizer. This contradicts the edge-only form of Theorem 3.2. □

4 Complexity of the local recognition problems

We measure deterministic time on a finite multi-stack Turing machine with binary input and output. A graph of order n is encoded by 1^n0 followed by its row-major adjacency matrix. A supplied binary chart records its two dimensions in the same unary encoding, followed by each vertex's binary coordinates in graph order. Supplied generators are listed as row-major permutation matrices in the same vertex order. In 3-CNF inputs, variable names may be binary integers of arbitrary length.

Theorem 4.1. *VIG recognition is* NP*-complete under polynomial-time many-one reductions, even on connected bipartite graphs of bounded maximum degree.*

Theorem 4.2. *EIG recognition is* NP*-complete under polynomial-time many-one reductions, even on connected vertex-transitive bipartite graphs without nontrivial modules. Every edge in every reduction output has an exchanging automorphism of order dividing four.*

For VIG, the reduction encodes a specified involutive exchange at the neighbors of a degree-two vertex. For EIG, it builds a transitive permutation action closed on ordered pairs and encodes a specified exchange as a graph-wide condition. A module is a set to which every outside vertex is adjacent either in full or in none. A nontrivial module has size between two and $|V|-1$. The reductions are proved in Appendix A.

5 Global compatibility and binary charts

An edge-colored graph has one color per edge, and its allowed automorphisms preserve the colors.

Definition 5.1. A *binary chart* identifies the vertices with $I \times L = \mathbb{F}_2^d \times \mathbb{F}_2^r$ and supplies the allowed permutation action

$$(x,u) \longmapsto (x+a, u+f(x)), \qquad a \in I, \quad f : I \to L.$$

Here a is the position shift and f the label profile. Write $T_a f(x) = f(x+a)$ and $\Delta_a = 1 + T_a$. Profiles can be represented by multilinear polynomials or by explicit value tables.

Theorem 5.2 (Fourth-root reduction)**.** *Let w encode a 3-CNF formula with s variables plus clauses. In deterministic time $2^{O((s+1)^{1/4})}(|w|+1)^{O(1)}$ one can construct a connected vertex-transitive bipartite simple graph G of order $2 \le n \le 2^{O((s+1)^{1/4})}$ such that G is RSG exactly when the formula is satisfiable. The graph has no nontrivial modules, its full automorphism group is a 2-group of nilpotency class at most four, and it admits compatible vertex and edge systems separately. The construction can supply a binary chart containing its full automorphism group and at most n generators for that group in the same time bound.*

The proof is in Appendix B. Truth values are stored in the quartic coefficients of a squarefree potential. The cubic parts of the vertex profiles are gradients of that potential. A false clause forces an element of the witness group that fixes both endpoints of the clause edge but fails to commute with any reversal of that edge. The construction first uses colors to encode the profile constraints and then removes them with Proposition B.1, which preserves Loos compatibility.

5.1 Recognition in a supplied chart

Theorem 5.3. *Let a connected graph with at least two vertices be supplied with a binary chart. For any compatible edge support H in the chart, write D for its position kernel. For every nonzero direction a of an edge,*

$$\operatorname{rank}(\Delta_a|_D) \le r.$$

Moreover $\dim D \le (d+1)r$*. If $r > 0$, its profiles have Boolean degree at most $1 + \lfloor \log_2 r \rfloor$. Compatible edge-only and full recognition in the chart take deterministic $2^{O((d+1)^2 r^2)} n^{O(1)}$ time. If the chart contains the full automorphism group, this is ordinary recognition.*

Proof. A compatible edge support projects onto I by transitivity. Its position kernel is an invariant binary space of profiles and is transitive on the labels of each block, because the position action is regular. At a crossing edge between blocks at x and $x+a$, write D_x, D_{x+a} for the kernels of evaluation at the two positions. A reversing witness acts on D by T_a, and so interchanges these subgroups. It centralizes $D_x \cap D_{x+a}$ by Theorem 3.2. On $D_x + D_{x+a}$ the image of Δ_a equals $\Delta_a D_x$. Thus the rank is at most

$$\dim\bigl(D_x/(D_x \cap D_{x+a})\bigr) + \dim\bigl(D/(D_x + D_{x+a})\bigr) = \dim(D/D_{x+a}) \le r.$$

The edge directions span I by connectivity. Choose a basis $a_1, \dots, a_d$ from the nonzero edge directions. Evaluation at zero and the d derivatives are jointly injective on D: a profile killed by every basis derivative is constant. The dimension bound follows.

For the degree bound, in these coordinates choose a nonzero coefficient of maximum degree $k \ge 1$ of a profile f. If its mask is J, the derivative in every direction of J is nonzero after all k derivatives. Fix $j \in J$. The 2^{k-1} elements

$$\left(\prod_{i \in U} \Delta_{a_i}\right) \Delta_{a_j} f, \qquad U \subseteq J \setminus \{j\},$$

are independent. In a nonzero dependence choose an inclusion-minimal U and apply the derivatives indexed by its complement in $J \setminus \{j\}$. The indicated term survives as the nonzero full derivative and all others vanish, since the derivatives commute and square to zero. All these elements lie in $\Delta_{a_j} D$, so $2^{k-1} \le r$. A binary linear coordinate change preserves the degree bound.

For the search, enumerate possible dimensions $m \le (d+1)r$. Enumerate for each basis direction a commuting matrix A_i of order dividing two on $\mathbb{F}_2^m$ with $\operatorname{rank}(A_i - 1) \le r$, by representing $A_i - 1$ as the product of an m-by-r and an r-by-m matrix. Enumerate an evaluation map $B : \mathbb{F}_2^m \to L$. Writing $x = \sum_i x_i a_i$, retain the profile spaces

$$\{\, x \longmapsto B A_1^{x_1} \cdots A_d^{x_d} z : z \in \mathbb{F}_2^m \,\}$$

for which the map from z is injective and the profiles are graph automorphisms. Test all basis profiles for the latter property. Test also that evaluation is onto at the root block. Every actual kernel is represented by this enumeration. The number of choices for fixed m is at most $2^{(2d+1)mr}$.

Fix a candidate D. Introduce an unknown chart lift g_i of each a_i through its label-profile table. For a prescribed shift, the allowed profile values at any two positions form an affine coset or an empty set: label translations on $L \times L$ taking the given pair's adjacency relation to the target pair's relation form a coset of its translation stabilizer, whenever a transporter exists. These sets can be found by enumerating the at most 2^{2r} pairs of translations and checking adjacency. For a pair with both positions equal, restrict to the common translation. Thus all graph-automorphism conditions on the basis lifts are linear constraints.

Require each lift to have square in D, and all lifts to commute modulo D. In the chart, if f_i is the profile of g_i, these requirements are $(1+T_{a_i})f_i \in D$ and $(1+T_{a_i})f_j + (1+T_{a_j})f_i \in D$. The subgroup generated by D and these lifts then has kernel precisely D: its quotient over D is generated by d commuting involutions and maps onto I. Let f_a be the profile of the ordered word in the lifts with shift a, linear in the lift variables. For each edge from (x, u) to $(x + a, v)$, introduce $h_e \in D$ and require

$$(1 + T_a)(f_a + h_e) = 0, \qquad f_a(x) + h_e(x) = u + v.$$

These equations impose an involution reversing the edge, including at $a = 0$. The pointwise endpoint stabilizer lies in D. It is centralized by a reversal exactly when every profile in D vanishing on the two endpoint positions is killed by Δ_a, a separate linear test on D. The resulting subgroup supports compatible edge witnesses by Theorem 3.2.

For full recognition, test the existence of a profile in D vanishing at one fixed root position and nonzero at the position of every neighbor. Enumerating D costs at most $2^{(d+1)r}$ per model. A root fixer lies in the abelian position kernel, so the profile is central there and gives the full system after the edge tests. Completeness follows by taking the kernel and lift data of a true support. Soundness follows from the subgroup criterion. All linear systems have size polynomial in the explicit graph order, giving the claimed time.

If $r = 0$, a full system is impossible because the chart acts regularly, and the edge problem is tested directly on the pure shifts. When $d = 0$, the position group is trivial, and the search uses no lift variables. □

Corollary 5.4. *Given a connected graph of order $n \geq 2$ and a binary chart containing its full automorphism group, RSG recognition takes $2^{O((\log n)^4)}$ time.*

Let $\mathcal{F}$ be the class of triples (G, χ, S) where G is connected, vertex-transitive and bipartite, has order $n \geq 2$, and has no nontrivial modules. Require that $\mathrm{Aut}(G)$ be a 2-group of nilpotency class at most four, that G admit compatible vertex and edge systems separately, that χ be a binary chart containing $\mathrm{Aut}(G)$, and that S generate $\mathrm{Aut}(G)$. For fixed $0 < \varepsilon < 4$, if RSG recognition on the supplied triples in $\mathcal{F}$ takes deterministic time $2^{O((\log n)^{4-\varepsilon})} n^{O(1)}$, then 3-SAT for a formula encoded by w, with s variables plus clauses, is decidable in deterministic time

$$2^{O((s+1)^{\max\{1/4,\,1-\varepsilon/4\}})}(|w| + 1)^{O(1)}.$$

The same transfer holds if the improved bound is guaranteed on triples in $\mathcal{F}$ with $|S| \leq n$.

Proof. The upper bound follows from Theorem 5.3 using $d + r = \log_2 n$. Each output triple of Theorem 5.2 belongs to $\mathcal{F}$, with $|S| \leq n$ and $\log_2 n = O((s+1)^{1/4})$. On these triples the recognizer takes time $2^{O((s+1)^{1-\varepsilon/4} + (s+1)^{1/4})}$. Adding the construction time gives the stated bound. □

In particular, a polynomial-time algorithm for unrestricted RSG recognition would give a deterministic 3-SAT algorithm with running time $2^{O((s+1)^{1/4})}(|w|+1)^{O(1)}$. The complexity of unrestricted RSG recognition remains open.

6 Primitive systems

Let G be connected with at least one edge, and write s_x for the vertex witness at x and r_e for the edge witness at e. A full system is *primitive* if its witness group H is primitive on the vertices, meaning that H is transitive and preserves no partition except the singleton and one-part partitions. Its *coarse type* is the vertex operation $x \triangleright y = s_x(y)$, up to isomorphism. In ordinary primitive quandle theory the group generated by the vertex reflections alone is required to be primitive [CS25, KS25]. Here the edge witnesses also contribute to H. For example, the group of vertex reflections on $\mathbb{F}_p^d$ with the operation $2x - y$ preserves every partition into cosets of a subspace.

For a nonabelian finite simple group T and a nonidentity involution $a \in \mathrm{Aut}(T)$, let $Q(T, a)$ be the operation

$$(tC) \triangleright (uC) = t\, a(t^{-1}u)C, \qquad C = C_T(a),$$

on the left cosets T/C. Here $C_T(a)$ is the fixed subgroup of a on T. Identifying T with its inner automorphisms, the coset tC corresponds to tat^{-1}, and the operation is conjugation on the class a^T. Powers of $Q(T, a)$ use the operation coordinatewise.

Theorem 6.1 (Primitive structural classification)**.** *The vertex displacement group $N = \langle s_x s_y : x, y \in V(G) \rangle$ is the unique minimal nontrivial normal subgroup of a primitive witness group H. For a root o, let θ be conjugation by s_o on N. The rooted vertex operation is determined by (N, θ). Its coarse type has one of the following three forms.*

1. *The affine operation $x \triangleright y = 2x - y$ on $\mathbb{F}_p^d$, where p is odd and $d \geq 1$.*
2. *The group core $x \triangleright y = xy^{-1}x$ on a nonabelian finite simple group T.*
3. *$Q(T, a)^k$ for T nonabelian finite simple and $k \geq 1$. If $k \geq 2$, some two points of a^T have trivial common stabilizer in T.*

The three forms are disjoint. In the third form, T and k are intrinsic and a is determined up to automorphism conjugacy after identifying T.

Proof. The subgroup N is normal in H by equivariance. If it were trivial, all s_x would be the same permutation, fixing every vertex. Let D be the subgroup of even words in all assigned involutions. It is normal of index at most two and it contains N. Along an edge $\{x, y\}$, the element $r_{\{x,y\}} s_x$ takes x to y, so D is transitive. Choose a minimal nontrivial H-normal subgroup $M \leq D$. It acts transitively by primitivity. Every s_x is then an M-conjugate of s_o, so all $s_x s_y$ lie in M. Thus $M = N$. Any other minimal nontrivial normal subgroup contained in D would likewise contain N. One meeting D trivially embeds in H/D, and its order would be at most two. This contradicts transitivity because G has at least three vertices. Thus $N = \mathrm{Soc}(H)$, the socle of H.

Write $s = s_o$. Its centralizer contains the maximal subgroup H_o. The permutation s is noncentral in H because a central element fixing a point of a transitive action fixes all points. It follows that $H_o = C_H(s)$. In particular the labels $x \mapsto s_x$ are injective. Since N is transitive, the rooted operation has the coset description

$$V(G) = N/C_N(\theta), \qquad (nC) \triangleright (mC) = n\, \theta(n^{-1}m)C, \qquad C = C_N(\theta).$$

A rooted operation isomorphism conjugates the intrinsic displacement group and the root reflection. It therefore induces an isomorphism of (N, θ), and the coset formula gives the converse.

A minimal normal subgroup of a finite group is characteristically simple. Hence N is elementary abelian or a direct power of a nonabelian finite simple group. In the abelian case, transitivity and faithfulness make its action regular. An invariant proper nonzero subspace for H_o would give a block partition, so H_o acts irreducibly. The fixed space of the nonidentity linear involution θ is H_o-invariant and must be zero. In characteristic two, $(\theta - I)^2 = 0$ would give a nonzero fixed space. Thus p is odd and $\theta = -I$, giving form 1.

If $N = T^\ell$, then H_o acts transitively on the simple factors because $H = NH_o$ and N acts on its factors by inner automorphisms. The permutation of these factors induced by s commutes with H_o. It either fixes every factor or exchanges them in pairs. In the first case the coordinate involutions are all isomorphic and nontrivial, giving $Q(T, a)^\ell$. In the second case, identify the two factors within each pair so that θ interchanges them. For each pair, the map $(u, v)\,\mathrm{diag}(T) \mapsto uv^{-1}$ identifies the quotient operation with the core on T. The collection of pairs is H-invariant because the factor permutation of s is central in the factor-permutation image of H.

An edge witness centralizes $N_x \cap N_y$ at its endpoints. If it moves a coordinate, this common stabilizer has no nonidentity element supported in that coordinate. For a core coordinate the common stabilizer at $1, z$ contains the diagonal of $C_T(z)$, which is nontrivial. Thus every edge witness preserves each core coordinate, as does each vertex witness, whose factor permutation only interchanges factors within pairs. Transitivity on the factors forces a single pair, giving form 2. In the fixed-factor case with $k \geq 2$, some edge witness moves a factor. Its two local endpoints must then form a two-point base for T. For such a pair (a, b), $ba \in T$ is nonidentity and has odd order. Otherwise its unique involutory power would centralize a, b and violate the base condition.

The intrinsic (N, θ) distinguishes the forms. In the nonabelian case it recovers the factor decomposition and the permutation of the factors by θ, and in the fixed case it recovers the coordinate involution up to automorphism conjugacy. □

6.1 The edge-orbit criterion

Take a candidate vertex operation $\mathcal{Q}$ of one of the forms in Theorem 6.1, in its coset model $N/C_N(\theta)$. The action of N is faithful. For an affine operation it is regular, and in the nonabelian cases the fixed subgroup contains no simple factor. The displacement group of the operation is indeed N. It is generated by left multiplications by $n\theta(n)^{-1}$ and is normal in N by equivariance. In the affine case these elements are all translations. In a core the elements from $n = (t, 1)$ project onto both factors. In a fixed-factor power there are nontrivial such elements in each simple factor. Normality then gives all of N. The vertex labels are injective because $s_{nC} = s_C$ holds exactly when $n \in C_N(\theta)$.

Put $A = \mathrm{Aut}(\mathcal{Q})$. The root stabilizer A_o identifies with $C_{\mathrm{Aut}(N)}(\theta)$ and $A = NA_o$. Indeed an operation automorphism fixing the root normalizes the displacement group and commutes with s_o. Its induced automorphism on N determines its action because N is transitive. Conversely an automorphism of N commuting with θ preserves the coset operation.

Proposition 6.2 (Edge-orbit realization)**.** *Let $\langle N, s_o\rangle \leq J \leq A$ act primitively. Consider graphs with vertex set $\mathcal{Q}$ and at least one edge. The full systems on these graphs with the prescribed vertex operation and witness group J are obtained by the following choices.*

1. *Select a nonempty collection of J-orbits of unordered pairs of distinct vertices.*
2. *At a representative $\{x, y\}$ of each selected orbit require $[s_x, s_y] \neq 1$, and choose an involution $r \in J$ exchanging x, y and centralizing $J_x \cap J_y$.*

3. *Require that the image of s_o and the images of all J-conjugates of the selected reversals generate J/N.*

The selected pairs form the edges, and conjugation transports their labels.

Proof. Injectivity and transport of the vertex labels give $s_x(y) = y$ if and only if $[s_x, s_y] = 1$. The pointwise endpoint stabilizer together with r generates the setwise stabilizer. Thus the centrality condition makes transport independent of the conjugating element, by Theorem 3.2. The graph is connected because its components form a J-invariant partition and it has an edge. The witness group contains the vertex displacement group N. All vertex labels are conjugate by N, so the listed images in J/N generate precisely the image of the witness group. This proves sufficiency and the generation test. Every primitive full system gives these same choices by taking its edge orbits and restrictions of its labelled action. □

For a pair with product $s_y s_x$ of odd order $m > 1$, the *odd midpoint reversal* is

$$r = (s_y s_x)^{(m+1)/2} s_x.$$

Conjugation by s_x inverts $s_y s_x$, so $r^2 = 1$ and $r s_x r^{-1} = s_y$. Injectivity of the labels implies that r exchanges the endpoints. A pointwise pair fixer commutes with both vertex labels and therefore with r.

6.2 Explicit realizations

Odd affine spaces. The affine operations of Theorem 6.1 are in the flat connected quandle classification [IT16]. To realize them as primitive full systems, let B be the signed permutation matrices on $V = \mathbb{F}_p^d$, with p odd. Take the Cayley graph with steps $\pm e_i$ and $\pm e_i \pm e_j$ for $i \neq j$. A block through zero for a regular translation group is an additive subgroup. For it to be a block for $J = V \rtimes B$ it must also be B-invariant. The independent sign changes project any nonzero vector of a B-invariant subspace onto a nonzero coordinate, and coordinate permutations give the other axes. Thus J is primitive. The vertex labels $z \mapsto 2x - z$ move every neighbor of x because p is odd.

For the edge $\{0, v\}$, with $v = e_i$, take k_v to negate the other coordinates. For $v = e_i + e_j$ use for k_v the swap of these coordinates, acting trivially elsewhere. In both cases use $r(z) = v - k_v z$. These two choices cover the step orbits that occur. Each k_v fixes v, squares to one, and centralizes B_v. In the first case B_v preserves the distinguished coordinate and acts by signed permutations on the complement. In the second it permutes the two active coordinates without signs and acts independently on the complement. Thus r reverses the edge and centralizes its pointwise stabilizer. The vertex labels generate all translations and $-I$. Together with the edge labels they generate the single-coordinate sign changes and the coordinate transpositions, since k_{e_i} times $-I$ changes just coordinate i. Proposition 6.2 now applies, including $d = 1$ where only the first step orbit is present.

Simple group cores. Write T^2 for the group acting on T by left and right multiplication, with root stabilizer the diagonal [SB25, Proposition 2.2]. This stabilizer is maximal. A larger subgroup containing the diagonal contains a nonidentity element in one factor after multiplication by an element of the diagonal. Conjugation by the diagonal and simplicity give that whole factor, and then both factors. Thus the left-right action is primitive. For the core take $J = \langle T^2, s_1 \rangle$, where s_1 is inversion on T. Join distinct x, y when yx^{-1} has odd order. This rule is invariant under the left-right action and inversion. It supplies an edge since a nonabelian simple group has order divisible by an odd prime. The product $s_y s_x$ has the same order as yx^{-1}: its left and right multiplication components have that common order, and the left-right action is faithful. Use the odd midpoint reversals. They are words in the vertex labels, which generate J, so the edge-orbit criterion gives a full primitive system.

An involution class and its powers. Let $T = A_5$, $\alpha = (1\,2)$, and $\beta = (1\,3)$, using these transpositions as automorphisms of A_5. The T-class of α is the set Y of two-element subsets of $\{1, \dots, 5\}$. The group S_5 is primitive on Y. Its stabilizer of $\{1, 2\}$ is $S_2 \times S_3$. Any element taking this pair to a pair meeting it in one point supplies a transposition crossing the two parts by conjugating $(1\,2)$. If it takes the pair to a disjoint pair, conjugating the complement's S_3 supplies a crossing transposition instead. The stabilizer and a crossing transposition generate S_5. At $k = 1$ take the graph on Y whose edges join intersecting distinct pairs. Use the transposition indexing a vertex as its label and the odd midpoint at an edge. The labels generate S_5, giving a primitive system on $Q(T, \alpha)$.

For $k \geq 2$, use the subgroup J of $S_5 \wr S_k$ on Y^k whose S_5 components all have the same sign. It contains $N = T^k$ and its quotient over N is the product of the common-sign group of order two with S_k. To check primitivity, put $o = (\alpha, \dots, \alpha)$ and $F = C_T(\alpha)$. A block through o corresponds to a subgroup U between F^k and T^k invariant under J_o: its points are the U-orbit of o, because each element of T^k carrying o into the block must preserve the block. Each coordinate projection P_i of U is invariant under $C_{S_5}(\alpha)$, which is realized diagonally in J_o. The product $P_i C_{S_5}(\alpha)$ is a subgroup of S_5 containing the maximal point stabilizer. Its intersection with T is P_i, since $C_{S_5}(\alpha) \cap T = F \leq P_i$. It follows that P_i is F or T. The coordinate permutations make these projections uniform. If they are all F, then $U = F^k$. If they are all T, conjugating the independently supported copy of $F \neq 1$ in each coordinate by U gives the full factor by simplicity. Hence $U = T^k$, completing the primitivity check.

Select the unordered pair orbit of $\{o, (\beta, \beta, \alpha, \dots, \alpha)\}$. Use $(m, m, \alpha, \dots, \alpha)(12)$ at this pair, where $m = (2\,3)$ is the local odd midpoint and the final (12) interchanges the first two coordinates. The local pointwise fixer of α, β in S_5 is $\langle (4\,5) \rangle$, so its intersection with T is trivial. A pointwise fixer of the displayed pair in J preserves the two active positions as a set. Its local maps there fix both α, β and coincide because their signs coincide. They commute with m. On the remaining positions the maps commute with α, so the reversal centralizes the pointwise fixer. It exchanges the endpoints and squares to one, and the vertex labels move the opposite endpoints because α, β do not commute. The root label supplies common odd sign, and conjugates of the reversal then supply all coordinate transpositions modulo N. The edge-orbit criterion gives the required full system on $Q(T, \alpha)^k$.

The determination of all admissible (T, a, k) in the involution-class form is a separate multiplicity problem. The structural theorem and the exact edge criterion above apply without a family-by-family list.

A Proofs of local hardness

A.1 A bounded-degree source for the vertex reduction

Lemma A.1. *A 3-CNF formula can be converted in polynomial time to a permutation 2-group P of nilpotency class at most two with an index-two subgroup Q such that $P \setminus Q$ contains an involution exactly when the formula is satisfiable.*

Proof. If the formula has no clauses, take $P = C_2$ in its regular action and $Q = 1$. The one element of $P \setminus Q$ is an involution. If the formula contains an empty clause, take $P = C_4 = \langle g \rangle$ in its regular action and $Q = \langle g^2 \rangle$. Both elements of $P \setminus Q$ have order four. In the remaining cases, repeat literals in clauses of length one or two to obtain an equivalent formula with three literals per clause.

For each clause $\ell_1 \vee \ell_2 \vee \ell_3$ of a 3SAT instance, introduce a fresh bit y and use the equations

$$y + (1 + \ell_1)(1 + \ell_2) = 0, \qquad y(1 + \ell_3) = 0$$

over $\mathbb{F}_2$, with a literal evaluated as its truth bit. Introduce a common bit t and homogenize every equation to degree two by multiplying linear terms by t and constants by t^2. The resulting

homogeneous quadratic equations $q_j(x,t) = 0$ with $t = 1$ are equivalent to satisfiability. The coordinates S_j used by each equation number at most four, including t, which is included in every S_j.

Choose a bilinear form b_j with $b_j(x,x) = q_j(x)$. On $L_j = \mathbb{F}_2^{S_j} \times \mathbb{F}_2$ define

$$(x,z)(w,h) = (x + w, z + h + b_j(x,w)).$$

Bilinearity gives associativity, and $(x,z)^2 = (0, q_j(x))$. Thus L_j has class at most two, exponent dividing four, and order at most 32.

Let P be the subgroup of the product of the L_j in which all copies of the same bit agree, represented by the regular actions of the L_j on disjoint sets. Add a two-point orbit on which the common bit t acts. All coordinates and all central flags are realized, and the permutation degree is polynomial. A generating set consists of one lift of each global bit basis vector and the independent central flags. The subgroup Q that fixes the two points individually has index two and is generated by omitting the lift of the t-basis vector. An element with $t = 1$ is an involution precisely when all of the q_j vanish. □

Proposition A.2. *It is* NP*-hard to decide whether a connected simple graph of bounded maximum degree admits an involution exchanging two specified nonadjacent vertices.*

Proof. Realize the permutation group from Lemma A.1 by a colored graph. For each local group L_j, make an element vertex for each $g \in L_j$, with a common color specific to j. For each chosen right-multiplication generator h of L_j, encode every arc $g \to gh$ as a path g, a, b, gh whose two interior colors encode j, h and the two distinct positions. An automorphism on the element vertices preserves these arcs if and only if it commutes with each right multiplication by h. Hence it is a left translation, determined by the image of the identity. Once the action on element vertices is fixed, these path vertices have unique images.

For each occurrence of a bit in a local group give it a private pair of port vertices, with a color identifying the occurrence but permitting the two vertices to be exchanged. Join each local element to the appropriate port value for every local bit. A left translation flips exactly the ports indicated by its bit coordinates. For each global bit, linearly order its occurrences and join consecutive pairs by the two equal-value edges. This enforces equal flips along the chain. Consequently the action on the whole colored graph is exactly the group P, acting on all the ports by their corresponding bits. Take the two vertices in any one port pair for t as the specified vertices. They are nonadjacent and their exchange by an involution is equivalent to the splitting test.

The number of local generators, local group order, and size of each local coordinate list are bounded. Each port has at most two synchronization neighbors, so the maximum degree is bounded. The t-chain, which runs through every local group, makes the graph connected. One may assume at least one clause, for example by adjoining a tautological clause.

To remove colors, index them by positive integers. At a vertex of color j, attach three pendant paths of lengths $1, 2, 3 + j$. The old vertices are exactly the vertices of degree at least three. The path lengths recover the colors, and the distinct lengths at each old vertex imply that every automorphism of the colored graph extends uniquely. The degree increases by three and the size remains polynomial. This gives the required ordinary graph. □

A.2 Vertex-involutive graph recognition

Duplicate each vertex of H into an independent pair, and replace each edge by the complete bipartite graph between its two pairs. Denote the resulting graph by $H^{\widetilde{2}}$, with vertices (i,x) for $i \in \{1,2\}$ and $x \in V(H)$. Two vertices are *open twins* if they have the same open neighborhood. Each open-twin class is independent, and between two distinct classes the adjacency is complete

or empty. Thus the quotient and its class sizes recover the graph. The quotient of $H^{\widetilde{2}}$ has the same classes as that of H, with each class size doubled.

Lemma A.3. *Let H be a graph without isolated vertices, and let u, v be distinct nonadjacent vertices. In $H^{\widetilde{2}}$, for each $i = 1, 2$ add $2\Delta(H) + 2$ independent vertices with neighborhood $\{(i, u), (i, v)\}$. Call these two sets the strips. The resulting graph $J(H; u, v)$ is a VIG if and only if H has an involution exchanging u, v.*

Proof. Suppose first that H has such an involution. Its simultaneous action in both layers, fixing the two strips pointwise, witnesses the vertex condition at each strip vertex. At a vertex over $x \notin \{u, v\}$, fix both copies of x, interchange the two copies of every other old vertex, and interchange the strips by a bijection and its inverse. This moves all the neighbors of the chosen vertex. At a vertex over u or v, fix all four vertices over u, v, interchange the copies of every other old vertex, and pair up the vertices inside each strip. Nonadjacency of u, v ensures that all old neighbors of each special vertex are moved. Each map described is an involution and preserves adjacency.

Conversely, the strips are exactly the open-twin classes of size greater than $2\Delta(H)$. Each open-twin class of H has size at most $\Delta(H)$, since it has a common neighbor. After the old vertices acquire their strip neighbors, two of them can be twins only if they were already twins in the blowup. No old vertex has the neighborhood of a strip vertex, since the old part of a nonempty old neighborhood contains both copies of each adjacent original vertex. The two strips have distinct neighborhoods. It follows that an involution witnessing the vertex condition at a strip vertex interchanges its two neighbors and preserves the old induced graph $H^{\widetilde{2}}$.

The restriction induces a permutation of order dividing two on the weighted open-twin quotient of the blowup. This quotient is the quotient of H with weights doubled, so we obtain a weight-preserving quotient automorphism that sends the class of u to the class of v. If the two classes coincide, the transposition of u, v is an automorphism of H. Otherwise, lift each quotient class transposition by a bijection and its inverse, choosing the bijection between the designated classes to send u to v. Fix each vertex in the fixed classes. The resulting permutation is an involution of H, since adjacency depends only on the classes. □

Proof of Theorem 4.1. By Proposition A.2, it is NP-hard to decide whether a connected bounded-degree graph H_0 has an involution exchanging two specified nonadjacent vertices u, v. Subdivide every edge once to obtain H. The old vertices form one side of the bipartition, so every automorphism of H exchanging u, v preserves that side and restricts to an automorphism of H_0. Conversely, an involution of H_0 exchanging u, v extends to H by its action on edges. Apply Lemma A.3. The resulting graph is connected and bipartite, because the two designated vertices lie on the same side of H. Its maximum degree is at most $4\Delta(H)+2$. The construction is polynomial, and membership in NP was checked above. □

A.3 Edge-involutive graph recognition

For Theorem 4.2, we reduce from 3SAT [Kar72]. The reduction uses a transitive permutation group A whose action is already closed on ordered pairs. The ordered-pair closure $A^{(2)}$ consists of the permutations that preserve each orbit of A on ordered pairs. We refer to those orbits as *orbitals*. We first construct A with two specified points exchanged by an involution precisely for satisfiable inputs. We then encode this single test in the global recognition problem.

A.3.1 A translation-invariant syndrome

Use the base position group $I = \mathbb{F}_2^{2k}$, with pairs of basis directions e_j, f_j for $1 \le j \le k$. Write $R = \mathbb{F}_2[I]$. Arrays on I are elements of R by using the group basis T^p for $p \in I$. Multiplication

by T^a translates the array by a. Put

$$x_j = 1 + T^{e_j}, \qquad y_j = 1 + T^{f_j}, \qquad \eta = \sum_{j=1}^{k} x_j y_j.$$

The ring has the monomial basis in x_j, y_j with exponents zero or one. Indeed these generators square to zero and the change from the group basis is invertible by expanding $T^{e_j} = 1 + x_j$ and $T^{f_j} = 1 + y_j$. The support S of η in the group basis has size at most $3k + 1$.

Let C consist of the positions that have exactly one set bit in each coordinate pair. For $p \in C$, let E_p be the monomial choosing x_j when pair j of p is $(1, 0)$ and y_j when it is $(0, 1)$. Given any binary data b_p on C, form the array z by expanding

$$z = \sum_{p \in C} b_p E_p$$

in the group basis. Write $z(p)$ for its coefficient at T^p. We call the elements of C *corners*. The following three properties will be used.

1. For each basis direction a, $(1 + T^a)z \in \eta R$. Multiplying E_p by the basis variable already chosen in its pair gives zero. Multiplying by its partner gives η times the monomial obtained by omitting that pair from E_p. Every other term of η has a repeated factor in this product.

2. Let $\pi : R \to \mathbb{F}_2^C$ keep just the coefficients of the monomials E_p. Then $\pi(\eta R) = 0$ because each surviving monomial of a product with η contains both variables from some pair. For a position q with a zero pair, $\pi(T^q) = 0$. For $q \in C$, $\pi(T^q)$ is the unit vector at q. Both assertions follow by expanding T^q pair by pair.

3. We have $z(p) = b_p$ at $p \in C$. At a noncorner position with no zero pair, $z(p) = 0$. In the expansion of each E_p, every pair is either zero or the corresponding pair of p.

Consequently, if an array o vanishes at noncorner positions with no zero pair and satisfies $o + z \in \eta R$, then $o(p) = b_p$ for every $p \in C$. Setting $o = z$ satisfies both requirements.

A.3.2 A uniform computation on the corner positions

We describe the binary fields and local tests used in the reduction. The number of fields and tests per position is $O(k)$, and the relations between fields at different positions use at most two positions each. Only the data b depend on the content of the source formula. The field and test layout can depend on its length.

The label w_p at position p includes a bit c common to all positions and a $2k$-bit address a_p. Impose the homogeneous linear pair relations $a_{p+d} = a_p + cd$ for all basis directions d and equality of the c bits. Pin $c = 1$ and $a_0 = 0$ at position zero, so $a_p = p$. Include an output bit $o(p)$ at every position. The local tests force $o(p) = 0$ at a noncorner position with no zero pair. They also check an accepting computation at the corner positions, whose initial input bits are $o(p)$ there. Positions with a zero pair can take arbitrary $o(p)$.

Here is one explicit implementation of the computation test. Fix a nondeterministic single-tape machine for 3SAT, reading a self-delimiting binary encoding of the formula on an initial input track. It starts at cell 1 just after a left boundary at cell 0. It can enter an accepting idle state only after returning to cell 1, and it has a polynomial time bound as a function of the encoding length. Fix powers of two for the time and tape dimensions, both polynomial in that length, large enough for its full running time, space, and the encoded input. Use at least two time values and at least four tape cells. Their binary addresses together

use $k = O(\log n)$ bits for input length n. At a corner, the second bit of each pair gives the corresponding bit of the time or tape address. Set b_p on the initial time row to the encoded input bits at their tape locations, padding with zeros, and set $b_p = 0$ on every other row. Boundary cells can have data zero and use their separate fixed boundary states.

A constant-size tuple of raw bits at a site records the tape symbol, the control state if the head is present, and a transition choice. The local predicate tests legal tile values and transition choices and checks the next time's tile at the same tape cell using the present tile and its two tape neighbors. For one head, a fixed transition choice at its cell determines its change of symbol, next control state, and head displacement. The same three tiles therefore determine the next symbol and head data at every site. Boundary cases reject moves out of range and ignore wraparound reads. Start-row constraints specify the unique initial head, the initial work state, and the input track using $o(p)$. The last-row test requires the accepting idle state at cell 1. The initial unique head and the update rule ensure a unique head in each successive row. Thus the local predicates accept exactly valid accepting tableaux on the prescribed input. Accepting runs can be padded by the accepting idle transition.

To read a neighboring numerical address, reserve a copy of its raw bits for each possible prefix XOR mask for binary increment or decrement in the relevant address coordinate. There are at most k masks per read. For each mask, a homogeneous linear pair constraint copies the raw bits from the corresponding translate. The translation toggles both coordinate bits in each pair indicated by the mask. An increment or decrement modulo the corresponding power of two uses one of these masks, determined by the trailing carry or borrow. Prefix tests and selection from the stored copies use $O(k)$ Boolean gates, since the tile alphabet and the number of reads are fixed. The tests for the corner set, for zero pairs, for time endpoints, and for boundary tape cells also use $O(k)$ gates. All of these translates preserve the corner set and its complement. Impose the tableau predicates only at corners and the output restriction only at noncorners with no zero pair.

Collect the raw bits, read copies, and Boolean gate values in the vector $w_p \in W = \mathbb{F}_2^{O(k)}$. Over $\mathbb{F}_2$ each gate equality can be homogenized using the marker c. For example, an AND with result bit r and input bits s, t uses $cr + st = 0$, and an XOR uses $c(r + s + t) = 0$. Use c for the constant one and compute NOT by XOR with c. Compute the guard and its predicate by gates before enforcing their implication, which is another Boolean condition of the same form. Write $Q(w_p) = 0$ for all these homogeneous quadratic tests and use the separate homogeneous output form $Q_o(w_p) = c\, o(p)$. The linear pair relations on w consist only of the address, marker and read copies described above. They are invariant under translation of positions with the labels unchanged. All gate values can be computed from the raw bits, read copies, address and output bit at the site. In particular, for an accepting tableau, all raw bits outside C may be chosen zero. Setting $o = z$ then gives a deterministic label w_* at zero depending only on $z(0)$ and the layout. The read copies at zero are all zero because their shifts preserve the complement of C.

A.3.3 The group on two copies of the positions

At a position use an ambient label group with coordinates (w, D, Y, Z). Here $w \in W$, D has a bit D^q for each $q \in S \cup \{0\}$, $Y \in \mathbb{Z}/4\mathbb{Z}$, and Z has one binary flag per coordinate of Q. A group product adds all coordinates, with the extra bilinear terms $B(w, w')$ in Z and $2B_o(w, w')$ in Y. Take binary bilinear maps with diagonal evaluations $B(w, w) = Q(w)$ and $B_o(w, w) = Q_o(w)$, by putting the quadratic coefficients in an upper triangular bilinear form. The terms in the Y coordinate are interpreted modulo four. Bilinearity gives associativity and both additional terms are central. The square of a label is zero in w, D, is $Q(w)$ in Z, and is $2(Y \bmod 2 + Q_o(w))$ in Y. The subset with $w = 0$ is central, with ordinary addition in the other coordinates.

The positions are now $(p, h) \in I \times \mathbb{F}_2$. Define the subgroup P of the product of the ambient label groups by these conditions.

- The coordinates w, Y, Z agree between halves at each p. Impose the linear pair relations on w from the computation construction. The tests $Q(w) = 0$ are reserved for the square of an eventual reverser.
- In each half h, impose $D^q_{p,h} = D^0_{p+q,h}$ and $Y_{p,h} \bmod 2 = \sum_{q \in S} D^q_{p,h}$. Thus writing $d_h(p) = D^0_{p,h}$, the parity of Y is the array ηd_h in either half. The tap arrays in the two halves can differ by a vector in the kernel of multiplication by η.

These conditions define a subgroup. The parity of Y and all coordinates w, D add linearly, and the two identical w entries add identical bilinear terms in Y, Z. The conditions also use at most two positions each. The purely local equality between the parity and its tap sum is allowed in the local projection. We can compute generators by solving the binary linear system on w, D and the parities of Y, lifting each vector of a basis to labels, and adding independent high bits of Y and Z at each p, in both halves together. All position projections are the same group L because the definition is invariant under ordinary position translations and the half swap. Act by left multiplication on copies of the regular L-set, one per position. The group L can be listed by enumerating the ambient labels and computing the indicated projection. Its size is $2^{O(k)}$.

For each basis direction a of I, compute by binary linear algebra an array d^a such that

$$(\eta d^a)(p) = z(p + a) + z(p).$$

This is possible by the first property of z. Interpret the binary values of z also as integers zero and one, and define a central local label D^a_p with tap field $(d^a(p + q))_{q \in S \cup \{0\}}$, $\mathbb{Z}/4\mathbb{Z}$ entry $z(p + a) - z(p)$ modulo four, and all other entries zero. These labels constitute a tuple in P when used in both halves together. Let U_p be the central label with $\mathbb{Z}/4\mathbb{Z}$ entry $2z(p)$ and all other coordinates zero. Its tuple used in both halves also belongs to P.

Adjoin the lifts σ_a of base shifts and the lift ρ of the half swap. From $(p, 0)$, σ_a sends a label unchanged to $(p + a, 0)$. From $(p, 1)$ it sends it to $(p + a, 1)$ by multiplying by D^a_p. The map ρ sends $(p, 0)$ to $(p, 1)$ without a label change and sends $(p, 1)$ to $(p, 0)$ by multiplying by U_p. All these maps normalize P because they permute positions by its invariances and use central label corrections. Let A be the group generated by these maps and P.

The position kernel of A is exactly P. The following computations verify all relations needed for the quotient on the given generators.

- The square of ρ is the tuple of U_p in both halves.
- In the square of σ_a, the $\mathbb{Z}/4\mathbb{Z}$ differences telescope to zero and the tap correction in half 1 comes from $d^a(p) + d^a(p + a)$. Its image under η is zero. Half 0 is unchanged. This lies in P.
- The products $\sigma_a\sigma_b$ and $\sigma_b\sigma_a$ give the same $\mathbb{Z}/4\mathbb{Z}$ correction in each half by telescoping. Their difference in half 1 is a tap field whose image under η is zero, since the two paths from p to $p + a + b$ give the same difference of z over $\mathbb{F}_2$. This difference lies in P.
- In comparing $\sigma_a\rho$ with $\rho\sigma_a$, the label correction of the former relative to the latter at target position $(p + a, 1)$ is D^a_p. At $(p + a, 0)$ its tap field is the same, since the taps have exponent two. Its $\mathbb{Z}/4\mathbb{Z}$ entry is $2z(p) - 2z(p + a) - (z(p + a) - z(p))$, which equals $z(p + a) - z(p)$ modulo four. Thus the relative correction at both positions is D^a_p. These translated taps and $\mathbb{Z}/4\mathbb{Z}$ fields give an element of P.

The quotient by P is therefore generated by $2k+1$ commuting involutions and has order at most 2^{2k+1}. It acts on the positions as the full regular translation group $I \times \mathbb{F}_2$ of that same order, so its action is faithful. The regular local actions and the transitive position action make A transitive. The construction takes polynomial time, with $2|I|$ positions and $2^{O(k)}$ labels per position.

A.3.4 Ordered-pair closure and the pinned exchange

The group A is closed on ordered pairs. Being in the same position is a union of ordered-pair orbitals, so a permutation preserving those orbitals preserves the position partition. Its induced permutation preserves every position difference, because each ordered-pair orbital has a single position difference. It therefore acts by a position translation. Compose with an actual lift of the induced position translation, leaving a permutation fixing every position. On each position this permutation preserves the local ordered-pair orbitals of the regular L-action. Indeed any element of A taking one point in a position to another point in the same position is in P. A regular action is closed on ordered pairs, so the local permutation is multiplication by an element of L. Applying the ordered-pair condition to the two identity labels in any two positions forces the multiplications to belong to the corresponding pair projection of P. The definition of P uses at most two positions at a time, so satisfaction of every pair projection implies membership in P. Undoing the lift proves $A^{(2)} = A$.

Pin the exchange between the identity label at $(0,0)$ and the label $(w_*, 0, 0, 0)$ at $(0,1)$. Here w_* is the label computed by setting the raw bits outside C to zero and evaluating the deterministic local gates at zero with $c = 1$, $a_0 = 0$, and $o(0) = z(0)$. It belongs to L even without a satisfying tableau. To see this, take the required address and marker fields everywhere, set raw bits and their reads zero everywhere, take the prescribed local auxiliary and output bits at zero, and choose all other gate and output bits arbitrarily. The local quadratic tests are not conditions defining P. All tap and central entries can be zero. The pinned exchange forces the position action of a witness to be that of ρ, so it is ρg for some $g \in P$. The local value of g at $(0,0)$ is the pinned label because ρ has no correction on passage from half 0 to half 1.

In the square of ρg, the w values cancel and the two Z flags at a fixed p sum to $Q(w_p)$. The $\mathbb{Z}/4\mathbb{Z}$ entry of this square at either half is

$$2(z(p) + (Y_p \bmod 2) + Q_o(w_p)).$$

These identities follow from agreement of w, Y, Z across the two halves, with one contribution U_p on each complete passage through them. The tap fields in the square sum across the two halves. Hence a witness with square one has $Q(w_p) = 0$ everywhere and $Q_o(w) + z \in \eta R$. The pin gives $c = 1$ and the actual address at every position. The quadratic tests force zero output at noncorners with no zero pair. Applying π gives $o(p) = b_p$ at every corner. The tableau predicates therefore certify acceptance of the encoded formula.

Conversely, take an accepting tableau and choose the unconstrained raw bits outside the corners to be zero. Set $o = z$ everywhere and evaluate the stipulated gates. Set both tap arrays, Y, and Z to zero in the tuple g. All linear conditions hold, its labels have the prescribed pin at zero, and the square formulas hold with zero flags because $Q(w) = 0$ and $Q_o(w) = z$. The tap part of the square is zero as well. Thus ρg has square one and exchanges the distinct pinned points. This proves the required equivalence with satisfiability.

The pin construction also gives an exchange whose fourth power is one on every input. Take any completion of w_* satisfying just the linear pair relations, and choose both tap arrays, Y, and Z zero. The local tests at zero hold because the gates there were evaluated and the tableau predicate is inactive. Thus $(\rho g)^2$ fixes the pinned vertices. Its coordinates everywhere

are central of order dividing two, by the square formula, so ρg exchanges the pinned pair and has fourth power one. Conjugation supplies such maps on the entire designated pair orbit.

A.3.5 From a closed transporter to a graph

Lemma A.4. *Given a transitive permutation group A by generators and two distinct points u, v, testing whether $A^{(2)}$ has an involution exchanging u, v polynomial-time many-one reduces to global EIG recognition on connected simple graphs. The output is vertex-transitive. If the designated pair has an exchange of order dividing four in $A^{(2)}$, every output edge has an exchange of order dividing four.*

Proof. **Removing colors for local edge witnesses.**

For a graph on b vertices with edge colors $1, \dots, s$, replace each vertex by a clique on the labels $\mathbb{Z}/q\mathbb{Z}$, where q is odd and $q > b(2s+1)$. Between fibers over a color-c edge, join labels i, j when $j - i \in \{-c, \dots, c\}$. There are no edges over a nonedge. In a clique meeting two or more fibers, choose a vertex in a different fiber from the one being counted. It has at most $2s+1$ neighbors there, so such a clique has size at most $b(2s+1)$. Thus the cliques of size q are exactly the fibers. The interspace degree $2c+1$ identifies its color. Every involution reversing a cross-fiber edge projects to an involution reversing the corresponding colored edge. Conversely, a color-preserving reverser of the base edge together with the common label reflection $z \mapsto i + j - z$ is an involution reversing the desired lifted edge. The label reflection alone handles edges within a fiber. Thus colored EIG with color-preserving witnesses reduces to ordinary EIG, preserving connectivity and vertex-transitivity. The construction is polynomial in the input size.

A fixed graph for distinguishing the two orientations.

Use the bipartite graph F on two copies of $\mathbb{Z}/13\mathbb{Z}$, with $(i, 0)$ adjacent to $(j, 1)$ for $j - i \in S = \{0, 1, 3\}$. The six ordered differences of distinct elements of S are distinct, so F has no 4-cycle. It is connected since $0, 1 \in S$. Every map $(i, b) \mapsto (c - i, 1 - b)$ is an involutive automorphism, with $c = i + j$ chosen to reverse any edge from $(i, 0)$ to $(j, 1)$. Its group is transitive by combining such a map with translations.

Let Ξ be the ordered orbital of $((0, 0), (1, 0))$ in the full group of F. It is distinct from Ξ^{T}, its transpose. To verify this, the graph of sharing a common neighbor on the zero side is the third power of the 13-cycle, with differences $\pm 1, \pm 2, \pm 3$. Its adjacent pairs at distances $1, 2, 3$ have respectively $4, 3, 2$ common neighbors, so every automorphism of F preserving that side preserves its original cycle. The zero-side action of any such automorphism is therefore a translation or a reflection. A reflection would take a right neighborhood, which is a translate of $-S$, to a translate of S. No reflection of the 13-cycle preserves S, since twice any element of S differs from the sum of the other two modulo 13. Thus S is not a translate of $-S$. Every side-preserving automorphism acts by translation on the zero side and cannot exchange 0 and 1 there. An automorphism exchanging those individual vertices must preserve the side, proving the claim about Ξ.

The colored graph.

Assume $N = |\Omega| \geq 3$, handling smaller cases directly. Set $U = \Omega \times \{0, 1\}$ and let $B = A \wr C_2$ act by independent copies of A and the copy swap. The ordered orbitals of B within the copies are the corresponding A orbitals merged across the two copies, and all ordered pairs between copies belong to one orbital. Let $Y = K_{N,N}$ on U have these cross pairs as edges. Every edge of Y is reversed by an involution in B, using an element of A carrying one endpoint's label to the other and its inverse in the other direction in the copy swap. The group $B^{(2)}$ is $A^{(2)} \wr C_2$: the cross orbital recovers the two halves, and the internal ordered orbitals give the respective restrictions. Let D be the unordered B-orbit of the pair $\{(u, 0), (v, 0)\}$.

Use vertex set $U^2 \times V(F)^2$, in that coordinate order. There are two geometry colors. One consists of the Cartesian edges in either of the first two coordinates along Y, with all other

coordinates fixed. The other consists of the Cartesian edges in either of the last two coordinates along F, with all other coordinates fixed. For every off-diagonal *ordered* orbital Δ of B, use a separate color E_Δ consisting of edges whose four ordered coordinate transitions form a tuple in

$$\Delta \times \Delta^{\mathsf{T}} \times \Xi \times \Xi^{\mathsf{T}}$$

or in the transpose of this product. The choice between a product and its transpose just picks an orientation of the same undirected edge. Distinct colors are disjoint because $\Xi \neq \Xi^{\mathsf{T}}$. Finally, use a designated color whose edges change one of the first two coordinates along a pair in D and one of the last two along an edge of F, fixing the other two coordinates. This color does not overlap the geometry colors or the colors E_Δ, which respectively change one and four coordinates.

The underlying graph is connected by its geometry edges and the independent group $B^2 \times \mathrm{Aut}(F)^2$ acts transitively by color-preserving automorphisms. Every geometry edge has a color-preserving reverser using an independent involution in B or $\mathrm{Aut}(F)$. An edge of E_Δ is reversed using a twisted swap on each coordinate pair. For instance, write its first two coordinate transitions as (x_1, x_1') and (x_2, x_2'). The ordered pairs (x_2', x_2) and (x_1, x_1') lie in the same orbital, so some $h \in B$ takes the former to the latter. The map $(x_1, x_2) \mapsto (h(x_2), h^{-1}(x_1))$ reverses the first two coordinates of the chosen edge and squares to the identity. Use the same construction with an automorphism of F on the last two coordinates. Swapping both coordinate pairs takes every orbital product to its transpose, and the independent maps preserve the respective ordered orbitals. The full involution thus preserves every color including the designated color.

Recovering the transporter.

Every color-preserving automorphism restricts on the geometry subgraph to a product of automorphisms of $Y \square Y$ and $F \square F$, because the components of each geometry color are exactly the fibers with the other coordinate pair fixed. In each Cartesian square a map preserves both coordinate directions or exchanges them globally. For an elementary verification, at a vertex of $Y \square Y$ two distinct incident edges are in the same direction if their other ends have N common neighbors, and are in different directions if those ends have two common neighbors. For $F \square F$, the corresponding counts are one and two because F has no 4-cycle. The local permutation of the two directions agrees at the two ends of every geometry edge, since both ends identify its direction and there are only two choices. Connectivity makes this permutation global. A direction-preserving map then acts independently on coordinates, by connectedness of every coordinate fiber.

Any reversal of a designated-color edge preserves the two coordinate directions inside each square, because only one of the two coordinates in each square changes on that edge. For the edge chosen to change the first and third coordinates, the reverser therefore acts separately on all four coordinates. In particular, its action on the third coordinate preserves the ordered orbital Ξ of F. For any ordered pair in a base orbital Δ, extend it to an edge of E_Δ with that ordered pair as its first transition and with a transition in Ξ in the third. The image still has its third transition in Ξ, so preservation of the color E_Δ forces the first transition of the image to lie in Δ. This holds for every Δ. Hence the first-coordinate map lies in $B^{(2)}$. On a designated edge using the internal pair $\{(u, 0), (v, 0)\}$, its restriction to copy zero is an involution of $A^{(2)}$ exchanging u, v.

Conversely, such an involution extends in one copy to $B^{(2)}$, and conjugating by B gives an involutive reverser for every pair in D. A $B^{(2)}$ action in a base coordinate preserves all the ordered orbital products. Combine it with a reversing involution for the moving F edge in the corresponding reference coordinate. This witnesses every designated-color edge. Thus the colored graph is EIG with color-preserving witnesses exactly when the input transporter exists in $A^{(2)}$. The orbit lists are obtained by orbit traversal on ordered pairs using the

input generators. The vertex set, adjacency and colors all have polynomial size. The color-removal construction completes the reduction to a simple graph. If the designated pair has an exchange whose fourth power is one in $A^{(2)}$, the same construction of designated-edge witnesses gives an order-dividing-four exchange for every designated edge. The other edges already have reversing involutions. The commuting reflection in the label blowup preserves this guarantee, and its global label translations together with the original transitive action preserve vertex-transitivity. □

A.3.6 Bipartite instances and completion of the reduction

Lemma A.5. *There is a polynomial-time many-one reduction from connected EIG recognition to EIG recognition on connected bipartite graphs without nontrivial modules. It preserves vertex-transitivity, and it preserves the property that every edge has an exchanging automorphism whose fourth power is the identity.*

Proof. For connected G of order $n \geq 2$, let $q = n + 1$ and use vertices (u, i, b) for $u \in V(G)$, $i \in [q]$ and $b \in \{0, 1\}$. For each u, connect all vertices over $(u, 0)$ to all over $(u, 1)$, forming a $K_{q,q}$. Over each edge uv of G, add the opposite-side edges with the same label i. These are the only edges. The graph is connected because the $K_{q,q}$ fibers are connected and their quotient is G.

The only copies of $K_{q,q}$ are the fibers. One side of any such copy contains two vertices over some u by pigeonhole. Any common neighbor of these two necessarily lies over u, since all relations to other fibers are matchings. The other side of the copy is therefore the entire opposite part over u, which forces the first side to be entirely over u as well. The fiber partition is preserved by all automorphisms and its quotient is G. A reversing involution for an edge between fibers projects to one reversing the corresponding edge of G. Conversely, an involutive reverser of uv acting on the first coordinate, together with the global side swap, reverses each edge over uv. A fiber edge from $(u, i, 0)$ to $(u, j, 1)$ is reversed by the side swap and the global exchange of i, j, with the identity on the label coordinate when $i = j$.

A module is a set M such that each vertex outside M is adjacent to all of M or to none of it. A proper module in a connected bipartite graph lies on one side. If it met both sides, no outside vertex could see all of it, so it would have no edge to its complement. A module of size at least two on one side would consist of vertices with identical open neighborhoods. In the constructed graph, vertices over different base vertices have different neighborhoods by the complete relation within their own fiber. Distinct labels on the same side over the same u have distinct neighbors in a fiber adjacent to u. This rules out all nontrivial modules. The output degree is $q + \deg_G(u)$ at a vertex over u. The reduction therefore preserves regularity. For a one-vertex input, use K_2, which has all the required properties. For the vertex-transitivity statement, use the automorphisms on the first coordinate together with global permutations of the labels and the global side swap. An edge exchange of order dividing four in the source lifts by the same commuting side swap used for involutions. Within each fiber the exchanges already have order two. □

Proof of Theorem 4.2. The construction in Sections A.3.1–A.3.4 reduce 3SAT to the designated exchange in a transitive action closed on ordered pairs, and supplies an exchange of order dividing four on every input. Lemma A.4 converts the test to global EIG recognition on a connected vertex-transitive simple graph with the same order-four guarantee on every edge. Lemma A.5 gives the bipartite restriction. These maps are deterministic and polynomial time. Proposition 3.3 gives membership in NP. □

B Proof of the fourth-root reduction

B.1 Compatibility-preserving color removal

Proposition B.1. *There is a deterministic polynomial-time color removal for connected edge-colored graphs of positive degree that preserves existence of a full system or a compatible edge system. The output automorphism group is the direct product of the input color-preserving automorphism group and a 2-group of nilpotency class at most two. A transitive color-preserving input group yields a vertex-transitive output.*

Proof. Let the base have $b \geq 2$ vertices and colors in $\{1, \ldots, c\}$. Set $W = \mathbb{F}_2^t$, $q = |W|$, and let $e_1, \ldots, e_t$ be its standard basis. Use $t \geq 3$, $M = t(t+5) + c$, and choose t large enough that

$$q > \max\{4bM,\ 2M(1 + \tbinom{M}{2} + t)\}.$$

Both $t = O(\log(b + c + 2))$ and polynomial q suffice, since the right side is polynomial in b, c, t.

We first construct sets $S_1, \ldots, S_c \subseteq W$ containing zero, with $|S_i| = t(t+5) + i$. For every S_i, the number of unordered pairs of its elements with difference e_j will be $j + 2$, and with any other nonzero difference at most two. For each j insert $j + 2$ pairs of form $\{a, a + e_j\}$, starting with a pair at $a = 0$. Then insert c singletons, using the first i of them with all pairs to make S_i. For each insertion choose a so no new point equals a previous point and no difference between a new point and a previous point equals a previous difference or any basis vector. This disallows at most $2M(1 + \binom{M}{2} + t)$ choices of a: each equation specifies a given the new point's offset, one old point, and a prohibited difference. Two new cross pairs at a single insertion can share a difference, but there are at most two such pairs for any difference because each new point determines its old partner from the difference. Later insertions never use that difference again. The only internal difference of a new planted pair is its specified basis vector. This proves the multiplicities. Deterministically try $a \in W$ at each insertion.

For each color i, the bipartite relation on two copies of W given by $x + y \in S_i$ has only the common translations as automorphisms preserving its two sides. Indeed, the count of common neighbors of two labels in one side is $|S_i \cap (S_i + d)|$, for their difference d. Each $d = e_j$ is characterized by the distinct count $2(j + 2) > 4$. A label permutation on that side therefore commutes with addition of each e_j and is a translation. This holds on both sides. Their translations must agree since S_i has no nonzero period. Its intersection count for every nonzero d is at most $2(t + 2) < |S_i|$.

The output graph has vertices (u, a, x) with u a base vertex, $a \in \mathbb{F}_2$, and $x \in W$. Join the two side fibers above a base vertex by all edges. Over a base edge of color i, join vertices (u, a, x) and (v, a, y) on the same side when $x + y \in S_i$. These edges and the internal ones make the graph connected.

The side fibers are intrinsic. Two distinct vertices in a side fiber have q common neighbors in the opposite side fiber. For vertices in different fibers, each of the $2b$ fibers contributes at most M common neighbors, since at least one vertex has only external neighbors there. Thus the common-neighbor count distinguishes the fibers. The complete relations identify the opposite side fibers over each base vertex, and their external degrees recover the colors. An automorphism induces a color-preserving base automorphism. Its side permutations must agree along every base edge because external edges join the same side. The local label actions are common translations along adjacent fibers on each side by the bipartite relation result, and agree throughout that side by connectivity. Internally the relation is complete, so the two sides can have independent translations. Conversely every described product action preserves the graph. Thus its full automorphism group is the direct product of the base color-preserving group and $W^2 \rtimes C_2$, with C_2 swapping the two copies of W. The latter group has nilpotency class at most two because all commutators lie in the central diagonal copy of W.

Suppose H supports a compatible base system. Use $H \times (W^2 \rtimes C_2)$ as supporting group upstairs. At a vertex use the base vertex involution with a nonzero label translation of the opposite side only. It moves all internal neighbors by the label action and all external neighbors by the base action. It is central in the vertex stabilizer, since the auxiliary vertex stabilizer consists of translations of the opposite side. An internal edge is reversed by swapping the label sides with the same translation on both, determined by the two endpoint labels. The auxiliary pointwise edge stabilizer is trivial, so the reverser is central in the setwise edge stabilizer. At an external edge use the base edge involution and the diagonal translation by the sum of the two labels. The base projection of its stabilizer is contained in the base edge stabilizer, and the diagonal translation is central in the auxiliary group. Thus the chosen involutions give the desired systems by the subgroup criterion.

Conversely, restrict to the layer with a fixed side and label zero. Every edge of the base occurs in that layer. A witness indexed in the layer preserves it, by the product form of the automorphisms and the requirement to fix its index or exchange the endpoints of its edge. The restrictions give the local conditions and all the conjugation identities. No valid restriction becomes trivial, because every base vertex has a neighbor and each edge has two distinct endpoints. □

B.2 Scalar derivative constraints

Let the pairing on a binary coordinate space be the usual dot product. The gradient of a squarefree polynomial is its vector of formal partial derivatives before evaluation on binary inputs.

Lemma B.2. *Let $f : \mathbb{F}_2^k \to \mathbb{F}_2^k$ satisfy that $a \cdot \Delta_a f(x)$ is constant in x for every a. Then $f = Q_\omega + Ax + c$, where Q_ω is the gradient of a homogeneous squarefree cubic ω, A is linear, and c is constant. Conversely every such map has the stated property, with constant $\omega(a) + a \cdot Aa$.*

Proof. Write every coordinate of f in its unique multilinear form. Suppose the greatest degree is $k_0 \geq 2$. Write $c_{i,U}$ for the coefficient on the monomial indexed by U in output coordinate i. For $|T| = k_0 - 1$ the coefficient on x^T in $a \cdot \Delta_a f(x)$ is

$$\sum_{j \notin T} \sum_i c_{i,T \cup \{j\}} a_i a_j.$$

It vanishes for all a. The linear coefficients force $c_{i,U} = 0$ when $i \in U$, and the quadratic coefficients force $c_{i,S\setminus\{i\}}$ to be common for all $i \in S$ at each (k_0+1)-set S. Call the common value γ_S. If $k_0 \geq 3$, choose $T \subset S$ of size $k_0 - 2$. In the coefficient of x^T, the monomial $a^{S\setminus T}$ has coefficient $3\gamma_S = \gamma_S$. Lower-degree terms contribute at most quadratic monomials in a there. This contradicts maximality unless $k_0 \leq 2$. At degree two the same coefficient conditions give precisely a cubic gradient. For a cubic monomial the terms containing an x-coordinate cancel in pairs in the scalar derivative, and its constant is the monomial evaluated at a. The affine contribution is $a \cdot Aa$, proving the converse. □

B.3 The reduction

Proof of Theorem 5.2. In time polynomial in $|w| + 1$, rename the variables consecutively, delete tautologies, remove repeated literals within each clause, and omit unused variables. If no clauses remain or the formula contains an empty clause, use the fixed output of the construction on x or $x \wedge \neg x$, respectively. Replace the k occurrences of a variable by distinct copies $x_1, \ldots, x_k$. For $k > 1$ add the binary clauses $\neg x_i \vee x_{i+1}$ cyclically. These force all copies to agree and add only linear size. Each copy occurs in at most three clauses, so the graph whose edges join variables in a common clause has maximum degree at most six. Greedy

seven-coloring gives distinct colors to the variables of every clause. Take $d \geq 3$ with d^4 at least the number of variables and $d = O((s+1)^{1/4})$. We describe one coordinate space for each of the seven colors, called a channel. Use identical coordinate spaces for all channels. The coefficient slots used by the variables can differ between channels.

The data potential. Use four coordinate groups $J'_1, \ldots, J'_4$. Each has a core C_i of dimension $d+1$, with d real coordinates and one tag coordinate. Use the tags of groups $1, 2, 3$ for polarity. Identify every core with the additive space of a field $F = \mathbb{F}_{2^{d+1}}$. In each channel put its variables in distinct ordered four-tuples of real coordinates, one from each group. Let W be a four-linear form on the groups. Its entries on all-real tuples are free. Fix the remaining entries as follows. For each unordered pair k, l of core groups, let i, j be the two complementary groups, in a fixed order. Allocate a new selector coordinate in group i and two new field copies in group j, disjoint from all other allocations. With the selector set to one, the two field inputs c, c' and core inputs $X \in C_k, Y \in C_l$ give the fixed slice

$$\mathrm{Tr}_{F/\mathbb{F}_2}(cXY + c'X^2Y).$$

These slices are the only fixed nonzero entries. In particular they use exactly two auxiliary coordinates outside the cores. The definitions specify W uniquely once its all-real entries are chosen. Put $J' = \bigoplus_i J'_i$.

Let $p = p_2 + p_3 + p_4$ be a binary squarefree polynomial on J', all of whose monomials use at most one coordinate per group. Its quartic part is W evaluated on the four groups. The cubic coefficients on triples containing a tag or a coordinate outside the cores are zero, except for the product of the three polarity tags, which has coefficient one. All-real cubic coefficients and all permitted quadratic coefficients can be arbitrary. Use the formal gradient $P = \nabla p$, with values in J', using the fixed coordinate pairing to identify the dual. For a squarefree homogeneous polynomial of degree j, its alternating j-form is the form with the same coefficients on unordered sets of distinct basis vectors. Denote the alternating form of p_4 by T. The third derivative of P in directions a, b, c is the constant vector with evaluation $T(-, a, b, c)$.

Two consequences of the fixed slices will be used. For a vector u in the cores with all four projections nonzero, let C_u be the span of the vectors $T(-, u, b, c)$ as b, c vary. It contains the full hyperplane $u^\perp$ of the *core* space. For one core pair k, l, insert its selector and vary the allotted field inputs in the other auxiliary group. In trace-dual coordinates the resulting pair of output spaces is the sum of

$$\{(cu_l, cu_k) : c \in F\}, \qquad \{((c'u_l)^{1/2}, c'u_k^2) : c' \in F\}.$$

Each has dimension $d+1$, and their intersection has dimension one. Indeed, equality requires $c = c'u_k$, and after squaring the first coordinate it requires $c' = 0$ or $c' = 1/(u_k^2 u_l)$. Their sum is the pair hyperplane annihilating (u_k, u_l). The pair hyperplanes span the full core hyperplane, since they contain the four within-group orthogonals and identify the nonzero scalar evaluations between groups. By alternation every element of C_u annihilates u. Furthermore C_u splits into its core hyperplane and a subspace on the auxiliary coordinates that is determined by the fixed slices alone. To see this, expand b, c into core and auxiliary parts. With exactly one auxiliary argument the output is auxiliary, and with zero or two the output is core. Thus the annihilator of C_u is spanned by the core functional $u \cdot -$ and a computable space of auxiliary-coordinate functionals.

The second consequence is that all third derivatives together span J'. To get each core output, insert a selector, a first field input, and a nonzero vector from the other core of that slice. For a selector output, insert a field input in its allotted group and two suitable core vectors. For the two field-copy outputs of a slice, insert its selector and the two core vectors. In

trace-dual coordinates the vectors (XY, X^2Y) span the two copies. A functional annihilating them would give $aX + bX^2 = 0$ for all $X \in F$, so $a = b = 0$. Each of these computations isolates its output group and the specified auxiliary allocation.

Two quadratic profiles. Pad J' by a space R of dimension $\dim J' + 3$ to obtain $J = J' \oplus R$. Identify J with the additive space of a binary field E. Write ϕ_2 for the homogeneous quadratic part of $x \mapsto x^3$ in the chosen binary basis. Its polar map is $B_\phi(x, y) = xy(x + y)$. For $x \neq 0$ the kernel in y is $\langle x \rangle$. Also the values $B_\phi(r, r')$ for $r, r' \in R$ span J. Otherwise some nonzero trace functional $\mathrm{Tr}_{E/\mathbb{F}_2}(\alpha -)$ would vanish on them. The radical of the alternating form $\mathrm{Tr}(\alpha B_\phi(x, y))$ consists of zero and the solutions of $x^3 = \alpha^{-1}$, as is seen by squaring the coefficient of y under the trace. It has dimension at most two. An isotropic subspace therefore has dimension at most $(\dim J + 2)/2$, which is strictly less than $\dim R$.

Let $U = J \oplus Z \oplus N$ with Z, N coordinate copies of J. Let V be a copy of U, and let $I = U \oplus V \oplus \langle w \rangle$. For $x \in I$ use coordinates (u, v, w), with $u = (j, z, n)$ in U. Extend P to I by ignoring all inputs off J'. For an alternating trilinear form ω, use Q_ω for the gradient of its squarefree cubic and B_ω for its polar map. Define Q_1, Q_2 from the cubics

$$w\, u \cdot v, \qquad z \cdot \phi_2(j),$$

respectively, with polar maps B_1, B_2. Let M be the coordinate projection of U onto J_1', and set

$$F_1(x) = Q_1(x) + (Mv, Mu, 0), \qquad F_2(x) = Q_2(x) + \iota_Z(n).$$

Here ι_Z denotes insertion into the Z coordinates of the output, and write A_0 for the linear part added to Q_1. For $a = (u, v, 1)$, the kernel of $B_1(a, -)$ is $\langle a \rangle$ and its image is $a^\perp$. The matrix on $U \oplus V$ is the invertible pairing of those two summands, and alternation gives the one-dimensional kernel. In every direction a, the Z-component of $B_2(a, b)$ is $B_\phi(a_J, b_J)$.

For a literal in a channel, let $j \in J'$ have its four variable coordinates set to one, and also its three polarity tags if the literal is negative. Define a position direction and its associated vector by

$$a = (u, 0, 1), \qquad u = (j, 0, \phi_2(j) + B_\phi(j, (1 + M)j)), \qquad b_a = ((1 + M)u, 0, 0). \tag{2}$$

Use the same formula with $j = 0$ in a channel absent from the clause. In either case $F_k(a) = B_k(a, b_a)$ for $k = 1, 2$. For $k = 1$ this follows from the V-component $(1 + M)u$. For $k = 2$ the Z-component is the stated polar value, and the other components vanish since both Z-inputs are zero.

The colored graph and the profile constraints. For every channel take two primary factors with position space I and label space $J' \oplus I \oplus I$, in that order for data and the two quadratic profiles. Take the product of all fourteen factors and two additional factors, each with a binary position and a binary label. Call the latter the marker and the parity factor. All edges are specified by position and label differences, so all additive translations of the vertices are color-preserving automorphisms.

The geometry colors in a primary factor shift only that factor, by one direction $(u, v, 1)$ with $u \cdot v = 1$. Use a separate color for every such direction and factor, with all label differences there allowed. These directions span I. For each basis vector e_i in U, take $(e_i, e_i, 1)$; adding $(e_i + e_j, e_i, 1)$ or $(e_i, e_i + e_j, 1)$ for a different j gives every pure coordinate direction on U or V. Then a direction with $w = 1$ supplies the last coordinate. The geometry of each extra factor shifts only its position and allows both label differences, in a dedicated color. A clause color shifts all seven first copies by their directions in (2), allowing all label differences in those copies and changing nothing elsewhere. Repeated clause direction tuples can be treated as one color.

For each channel and every nonzero $a \in I$, comparison edges shift both its copies by a and the marker position by one. Give these edges the color of the tuple of the two values $a \cdot (\delta_k + \delta'_k)$, $k = 1, 2$, where δ_k, δ'_k are their quadratic label differences. When $a = e_i$ is a basis vector of J', include in the tuple the i-th coordinate of the sum of the two data label differences. These relations allow all label differences in the shifted factors. Every type of position shift and geometry used in the construction has a disjoint color set.

Guard edges shift a single primary factor by a nonzero a and the marker position by one. They also shift the parity position by one if $w(a) = 1$. Their color records a tuple of rows of the form $\ell(\delta) + c\,\delta_h$, where δ lists the three primary label differences and δ_h is the marker label difference. All label differences in the shifted factors are allowed. For each such position shift use one tuple with all requested rows. This avoids any parallel colored edges. All other factors remain unchanged. The rows will be specified below using the notation $\ell(\Delta_a f(x)) = c$, where $\Delta_a f(x) = f(x + a) + f(x)$. This denotes exactly the stated guard color rule. The marker makes the equation applicable to a root witness, as explained next.

The geometry colors recover the factor coordinates. For a set of colors dedicated to one factor, its connected components are precisely the fibers obtained by fixing all other factors, by the spanning property and the free label differences. Preservation of all these fiber partitions makes an automorphism coordinatewise. Within a primary factor, the direction colors identify the positions as the classes of equal color-neighbor sets, and on those positions each automorphism commutes with all the named geometry shifts. Thus it is a position translation. The comparison colors force each label permutation to be a translation. In fact, fix the other three endpoint labels in a comparison, and vary the fourth arbitrarily. Each selected row then changes by a constant under its label permutation. These rows range over a spanning set of duals of the primary label space. The extra factors have this form automatically, since their positions and labels are binary. In particular the entire position action is regular. A vertex involution fixing the global zero has zero position shift and zero value there in each label profile. If it moves all neighbors, its marker label profile has values $0, 1$ at positions $0, 1$, by the marker geometry. This gives the claimed guard equations.

Since the two positions of a comparison vary independently, $a \cdot \Delta_a f_k(x)$ is constant in x for each quadratic label profile of a root witness. The calculation in Proposition B.2 gives $f_k(x) = Q_{\omega_k}(x) + A_k x$. In particular its scalar offset is the cubic of ω_k plus $a \cdot A_k a$. Include the guard rows $a \cdot \Delta_a f_k = a \cdot F_k(a)$ for every nonzero a. They fix $Q_{\omega_k} = Q_k$ and the quadratic and diagonal parts of the scalar linear offset. For the second profile also require zero derivative in every unit direction in every output coordinate outside $J \oplus Z$. Its prescribed quadratic part is zero in those outputs. Thus f_2 takes values only in $J \oplus Z$.

The first profile has $A_1 - A_0$ alternating. For a unit direction e_j and output row e_i^* annihilating $B_1(e_j, -)$, require that the offset of that row is its value for F_1. The only remaining entries of the alternating difference are the pairs $(u_i, v_i), (w, u_i), (w, v_i)$. For direction $e_{u_i} + e_{u_j}$ with $i \neq j$, the row $e_{v_i}^* + e_{v_j}^*$ annihilates the polar image. Again require its offset for F_1. This makes all remaining (u_i, v_i) entries equal. These freedoms are exactly the maps $B_1(c, -)$ for $c \in I$, as seen by inserting the three types of coordinate vectors into $w\,u \cdot v$. Consequently each first profile of an actual root witness has form

$$f_1(x) = F_1(x) + B_1(c, x). \tag{3}$$

For the data profile D, require $\Delta_{e_j} D_i = 0$ if i, j are in the same group, including $i = j$, or if e_j is outside J'. For distinct data coordinates i, j also require $(e_i + e_j) \cdot \Delta_{e_i + e_j} D = 0$. Because D_i ignores its own input, this is $\Delta_{e_j} D_i = \Delta_{e_i} D_j$. In squarefree coefficients these equations make the coefficients with output index i and input set $S \setminus \{i\}$ agree across all $i \in S$. They are zero whenever S uses two coordinates from the same group. This proves that D is the gradient of a potential as above, initially with undetermined cubic and quartic coefficients.

For every triple of three distinct groups containing a tag or an auxiliary coordinate, let t be its indicator in J', and use the position direction $a = (t, 0, 1)$, viewing t as lying in U. The expression $t \cdot \Delta_a D(x)$ is the cubic coefficient at the triple plus the linear functional obtained by contracting the unknown alternating quartic form on the triple. This follows by expanding the gradient of each monomial. A term with h factors supplied by the direction in that expansion occurs h times, so the odd counts give the cubic constant and the quartic linear term. Write θ for the linear functional from the *prescribed* quartic slice on this triple. It annihilates a. Since $w(a) = 1$, choose a row ρ with $\rho B_1(a, x) = \theta(x)$, and impose the guard

$$t \cdot \Delta_a D(x) + \rho\, \Delta_a f_1(x) = p_3^{\text{fixed}}(t) + \rho F_1(a).$$

Comparing the linear terms fixes every quartic entry involving a tag or an auxiliary coordinate. The constant part fixes the indicated cubic coefficient when $f_1 = F_1$. More generally replacing the three profiles by their translates $f(x + c) + f(c)$, with c from (3), preserves every guard and gives $f_1 = F_1$. Thus after this normalization the data profile is P with exactly the stated coefficient constraints.

Finally, for every clause direction a impose two types of joint rows in its first-copy factor. For every output row ℓ of the second quadratic profile, choose ρ with $\rho B_1(a, x) = \ell B_2(a, x)$ and impose $\ell \Delta_a f_2 + \rho \Delta_a f_1 = 0$. Such ρ exists since the functional annihilates a, and the canonical profiles meet this offset by (2). For an active channel, take a basis of the auxiliary rows annihilating C_j, where $j = a_{J'}$. For each such ℓ, let θ be the linear part of $\ell \Delta_a P(x)$ for the prescribed quartic slices and cubic tag term. The quadratic part vanishes by the choice of ℓ. The linear part is the same for every normalized allowed P, since free cubic and quartic coefficients use only real coordinates. The functional annihilates a because the derivative is invariant under translation by a. Also $\theta(b_a) = 0$ and $\ell P_0(a) = 0$ when P_0 uses only the fixed coefficients. Indeed the auxiliary outputs of the quartic gradient need at least one auxiliary input, and both a and b_a have only core inputs on J'. Thus we can impose $\ell \Delta_a D + \rho \Delta_a f_1 = 0$ with $\rho B_1(a, x) = \theta(x)$. All canonical profiles with zero quadratic coefficients meet these offsets. For a normalized actual profile the equations give

$$f_2(a) = B_2(a, b_a), \qquad \ell\big(D(a) + \Delta_{b_a} \Delta_a D(0)\big) = 0 \tag{4}$$

for every listed auxiliary row. For the first equality, evaluate the guard at zero and use $F_1(a) = B_1(a, b_a)$. For the second, the two terms both evaluate to $\theta(b_a)$. In an inactive channel $a_{J'} = 0$, so the data profile has zero derivative in direction a.

Reading a clause. In a normalized active channel put $j = a_{J'}$. The vector $D(a) + \Delta_{b_a} \Delta_a D(0)$ belongs to C_j exactly when its pairing with j vanishes, by (4) and the description of C_j. This scalar is

$$W(j_1, j_2, j_3, j_4) + p_3(j). \tag{5}$$

The quadratic part of the potential has alternating gradient matrix, so contributes zero. For the cubic part, the polar term in the double derivative vanishes under pairing with j, by alternation, and the gradient at j yields $p_3(j)$. For the quartic part, its double derivative plus its value at j is the sum of the gradient at Mj and $(1 + M)j$. The first vanishes, and the second yields precisely the contraction on the fourth missing group, namely group 1. The value of W in (5) is the variable's coefficient; all fixed entries with a tag and no auxiliaries are zero. The value of $p_3(j)$ is the sum of the four real cubic coefficients on its variable coordinates, plus the polarity bit. For each variable define its assignment bit as the quartic coefficient plus those four real cubic coefficients. Consequently (5) is the literal value, with either sign and independently of other variables in that clause. Every assignment is obtained by taking the free cubic coefficients zero and setting the corresponding quartic coefficients.

Necessity of satisfiability. Let a full system exist, and let H be its generated group. It is transitive by the edge witnesses, hence projects onto the whole regular position action. Its position kernel is abelian. If s is its root vertex witness, conjugating by a lift of a position shift in a single factor and multiplying by s gives exactly the derivative of that factor's profile, acting trivially elsewhere. Iteration gives all higher derivatives, without any contribution from the label corrections in the shift lifts.

In each primary factor H contains the pure constant profiles needed to normalize these derivatives. The third derivatives give all pure data constants by the spanning calculation for W. To obtain the first quadratic label constants, take double derivatives in directions w and a basis vector of U or V, or in a pair of matched U, V directions. Their data and second quadratic outputs are zero and their first quadratic outputs span I. For the second quadratic constants on Z, double derivatives in the padding R give the spanning values $B_\phi(R, R)$, with the other profiles unaffected. For constants on J, use shifts in Z and J. Their J-values span since a functional annihilating them would be a vector in the common radical of B_ϕ, which is zero. Their other outputs vanish. The second profile has no values off $J \oplus Z$. This proves the normalization claim for every profile of s, including arbitrary linear parts allowed in that profile.

Use these constants and first derivatives to replace each primary profile of s by $f(x+c)+f(c)$ inside H. This leaves its marker and parity profiles unchanged and gives the normalized profiles above. The resulting element is used only as an element of H. Suppose a clause is false on the extracted assignment. Use the clause edge from zero to its position tuple, with both label tuples zero. Add to the normalized element the normalized first derivative in direction b_a of each participating primary factor. At the other endpoint the first quadratic value is zero by (2). The second quadratic value is zero by (4). The data value belongs to C_j because the literal tests vanish. It can be canceled by normalized double derivatives in that factor, whose evaluation on the direction a spans C_j and whose normalized quadratic profiles vanish. In inactive channels the data value is already zero. Thus we obtain an element of H fixing both endpoints. It fails to commute with every reverser of this edge in H. Any such reverser has the edge position shift, and conjugation of the position kernel only translates its profiles. The derivative of the constructed first quadratic profile along an active clause direction retains the nonzero linear part $B_1(a, -)$. This contradicts centrality of the assigned edge involution in its setwise stabilizer.

Sufficiency. For a satisfying assignment use the canonical quadratic profiles, the prescribed quartic gradient with its variable coefficients, the fixed cubic tag term and no other cubic or quadratic coefficients. Use these in both copies, and the identity position-to-label profile in each binary extra factor. The assembled profile s is an involution fixing the global zero and satisfies all comparisons and guards. It moves every root neighbor. For a spanning geometry direction, the first quadratic scalar $a \cdot F_1(a) = 1$. For a clause direction in an active channel its first quadratic V-value $(1 + M)u$ is nonzero, since the variable uses the other three groups. All other edges shift a binary extra position where the profile is nonzero.

Use the support generated by s and all additive translations of the vertices. Its position kernel is abelian, so s is central in the root stabilizer. The kernel is spanned by the constant profiles, s, and the successive derivatives of its primary profiles. In particular any kernel element can be written as εs plus constants and, in each primary factor, a first derivative in one direction b and a sum of higher derivatives of that factor's profile. This follows from the derivative identity $\Delta_b + \Delta_{b'} = \Delta_{b+b'} + \Delta_b \Delta_{b'}$.

For an edge take the additive translation by its endpoint difference. The pointwise edge stabilizer belongs to the position kernel. Consider such an element, in the above form. A change of marker or parity position forces $\varepsilon = 0$. On a spanning geometry edge this follows

instead from $a \cdot F_1(a) = 1$ and $a \cdot B_1(a, b) = 0$. If $\varepsilon = 1$ on a clause edge, the first quadratic endpoint tests give $B_1(a, b) = F_1(a)$ in each first-copy channel. Therefore $b = b_a$ modulo $\langle a \rangle$. The normalized data difference at the endpoints, paired with j, is (5), since the higher derivatives contribute elements of C_j. The first derivative contribution to this comparison is unchanged by adding a to b, because $\Delta_a^2 = 0$. These calculations can use the root edge, by translation. At least one literal is true, so again $\varepsilon = 0$.

With $\varepsilon = 0$, the derivatives of the quadratic profiles along any factor shift a are constant, and must vanish since this element fixes both endpoints. In particular the second quadratic profile's Z-value gives $B_\phi(a_J, b_J) = 0$. If $a_J = 0$, the data derivative vanishes. Otherwise $b_J \in \langle a_J \rangle$, so the first derivative of the data profile in direction b has zero derivative along a. The higher data derivatives have degree at most one and give a constant derivative along a. It vanishes by the endpoint tests. The extra profiles are constant when $\varepsilon = 0$. Hence the edge reverser centralizes its pointwise stabilizer, and also its setwise stabilizer, generated by the pointwise stabilizer and the reverser. Choose these reversers at edge orbit representatives and transport them by the subgroup criterion. A transported witness can differ from the pure translation at its edge.

Size and additional properties. Every position and label dimension is $O(d)$, with a constant number of factors. All fields, rows and directions can be computed in time $2^{O(d)}$. For example, construct each binary field by enumerating monic polynomials of the required degree and rejecting those divisible by a monic polynomial of positive degree at most half as large. Use binary elimination to choose each row. In particular C_j's auxiliary part uses only fixed coefficients. The colored graph has $2^{O(d)}$ vertices and colors and can be built explicitly within that bound. It is connected by the geometries. It is bipartite by the sum of the primary w-coordinates and the two binary positions. Clause edges change all seven first-copy w's, comparison edges change matched directions and the marker, and the parity adjustment makes every guard cross the bipartition.

Every automorphism of the colored graph has an additive position shift and label profiles of degree at most three. Indeed remove its position shift by an additive translation. Its quadratic profiles obey the comparison conditions, which give degree at most two without the marker condition. For the data conditions above the guard constants were zero, so they give the same gradient representation for this automorphism, up to constants. Thus the automorphism group is a subgroup of the semidirect product of an elementary abelian group of profiles of degree at most three by the elementary abelian position translations. Commutators with position translations lower polynomial degree by one, so this group is a 2-group of class at most four. The graph always has a compatible edge system using only the translations. The same canonical root profile also always moves all neighbors regardless of the assignment, so with translations it gives a compatible vertex system.

Apply Proposition B.1 to remove the colors. Its proof gives a direct product with an auxiliary 2-group of class at most two, and preserves the two separate systems by the same product formulas. Its output bipartition is the sum of the base side and the auxiliary side. The output neighborhoods are distinct. In different side fibers their intersection is smaller than the size of a single neighborhood, by the common-neighbor bound of the color removal. In one side fiber their external neighborhoods over an incident base edge are distinct, by nonperiodicity of the label set for that color. In a connected bipartite graph, a proper module lies on one side, since an edge from the module to its complement rules out meeting both sides. A module on one side consists of vertices with the same neighborhoods. Thus no nontrivial modules occur.

The output is vertex-transitive and has at least two vertices. Its order and the time to construct it after preprocessing are bounded by $2^{O(d)}$. The auxiliary side and field labels of the color removal are one additional binary position and label factor, with regular side shift and

additive local labels. Together with the colored factors this gives a single chart with binary position translations and binary label translations containing the full automorphism group. Full automorphism generators are obtained by binary elimination on the label profile equations of the colored graph, adjoining the position translations and the generators of the auxiliary factor. Indeed each zero position shift preserves the colors exactly when its changes in the specified rows vanish at every pair of positions. This is a linear system of size polynomial in the explicit graph order. In the output chart with position dimension d_0 and label dimension r_0, the position kernel has a basis of at most $r_0 2^{d_0}$ profiles, and d_0 shifts generate the position group. The list therefore needs at most $r_0 2^{d_0} + d_0 \leq 2^{d_0+r_0} - 1 = n - 1$ generators. Writing the graph, chart, and permutation matrices takes time polynomial in n, which fits the bound after preprocessing. □

C Finite verification and reproducibility

Local hardness. The first listing constructs the binary syndrome ideal in the group basis and checks the corner projection and shift relations for every corner-basis input with $1 \leq k \leq 4$ in Appendix A. It also checks the common-neighbor counts and all candidate side-preserving dihedral maps for the 26-vertex orientation detector. These regressions cover the stated finite inputs. Verification of the general tableau simulation, pinned exchange, and graph transfers uses the proofs.

Global compatibility. The second listing checks field slices and polar radicals over $\mathbb{F}_{16}$ and $\mathbb{F}_{32}$, scalar comparisons, geometry shifts, the two auxiliary profiles, and quartic clause and triple-guard identities. The field arithmetic is specified by irreducible binary polynomials in the listing. The tests enumerate coefficient or direction choices in the indicated small models. They exercise algebraic ingredients of Appendix B. An independent end-to-end check of the entire fourth-root construction remains open.

Chart recognition and primitive realizations. The algorithm of Theorem 5.3 specifies the kernel enumeration and the affine systems for the lifts. Its input is an explicit graph and chart. The primitive realizations in Section 6.2 use direct stabilizer and generation proofs.

Running the listings. Copy each block between its verbatim delimiters from this LaTeX source into a `.py` file. Run each file with Python 3.10 or later. Both programs use only the standard library. Each assertion checks a stated finite identity, and each program prints the families of tests it completed.

C.1 EIG syndrome and orientation checks

```
"""Finite regressions for the syndrome and orientation detector in the EIG proof."""
from itertools import product


def row_basis(columns):
    basis = {}
    for column in columns:
        while column:
            pivot = column.bit_length() - 1
            if pivot not in basis:
                basis[pivot] = column
                break
            column ^= basis[pivot]
    return basis
```

```
def in_span(vector, basis):
    while vector:
        pivot = vector.bit_length() - 1
        if pivot not in basis:
            return False
        vector ^= basis[pivot]
    return True


def submasks(mask):
    current = mask
    while current:
        yield current
        current = (current - 1) & mask
    yield 0


def check_syndrome(k):
    size = 1 << (2 * k)
    integers = range(size)
    corners = [
        sum(1 << (2 * j + b) for j, b in enumerate(choices))
        for choices in product((0, 1), repeat=k)
    ]
    eta_support = set()
    for j in range(k):
        eta_support.symmetric_difference_update(submasks(3 << (2 * j)))
    basis = row_basis(
        sum(1 << (p ^ q) for q in eta_support)
        for p in integers
    )

    def project(v):
        result = 0
        for p in integers:
            if v >> p & 1:
                for i, corner in enumerate(corners):
                    if p & corner == corner:
                        result ^= 1 << i
        return result

    for p in integers:
        column = sum(1 << (p ^ q) for q in eta_support)
        assert project(column) == 0
        if any((p >> (2 * j)) & 3 == 0 for j in range(k)):
            assert project(1 << p) == 0
    for i, p in enumerate(corners):
        z = sum(1 << q for q in submasks(p))
        assert project(z) == 1 << i
        assert [q for q in corners if z >> q & 1] == [p]
        for j in range(2 * k):
            shifted = sum(1 << (q ^ (1 << j)) for q in submasks(p))
            assert in_span(z ^ shifted, basis)
    print(f"k={k}: {len(corners)} corner columns and rank {len(basis)} verified")


def check_orientation_detector():
    n = 13
    steps = {0, 1, 3}
    neighborhoods = [{(i + s) % n for s in steps} for i in range(n)]
    right_neighborhoods = {frozenset((j - s) % n for s in steps) for j in range(n)}
    assert max(len(neighborhoods[i] & neighborhoods[j])
```

```
            for i in range(n) for j in range(i)) == 1
    proximity = [{j for j in range(n) if j != i and neighborhoods[j] & neighborhoods[i]}
                 for i in range(n)]
    assert [len(proximity[0] & proximity[d]) for d in (1, 2, 3)] == [4, 3, 2]
    preserving = set()
    for sign, shift in product((-1, 1), range(n)):
        mapped = {frozenset((sign * i + shift) % n for i in star)
                  for star in right_neighborhoods}
        if mapped == right_neighborhoods:
            preserving.add((sign, shift))
    assert preserving == {(1, shift) for shift in range(n)}
    for shift in range(n):
        for i, j in product(range(n), repeat=2):
            assert ((j - i) % n in steps) == (
                ((shift - i) - (shift - j)) % n in steps
            )
    print("26 candidate side-preserving maps and all reflected edges verified")


if __name__ == "__main__":
    for dim in range(1, 5):
        check_syndrome(dim)
    check_orientation_detector()
```

C.2 Fourth-root algebraic checks

```
"""Finite and symbolic regression checks for the fourth-root RSG construction.

The checks exercise identities on small coordinate models. They do not
instantiate the entire reduction or certify its all-input proof.
"""

from itertools import combinations, product


def bits(mask):
    while mask:
        bit = mask & -mask
        yield bit.bit_length() - 1
        mask ^= bit


def mask_of(indices):
    return sum(1 << index for index in indices)


def dot(left, right):
    return (left & right).bit_count() % 2


def row_basis(rows):
    basis = {}
    for row in rows:
        while row:
            pivot = row.bit_length() - 1
            if pivot not in basis:
                basis[pivot] = row
                break
            row ^= basis[pivot]
    return basis


def annihilator(rows, coordinate_mask):
```

```
    basis = row_basis(row & coordinate_mask for row in rows)
    for free in bits(coordinate_mask):
        if free not in basis:
            solution = 1 << free
            for pivot, row in sorted(basis.items()):
                if dot(row, solution):
                    solution ^= 1 << pivot
            yield solution


class Field:
    def __init__(self, degree, modulus):
        self.degree = degree
        self.size = 1 << degree
        self.modulus = modulus
        assert all(self.power(x, self.size - 1) == 1 for x in range(1, self.size))

    def multiply(self, left, right):
        result = 0
        while right:
            if right & 1:
                result ^= left
            right >>= 1
            left <<= 1
            if left & self.size:
                left ^= self.modulus
        return result

    def power(self, value, exponent):
        result = 1
        while exponent:
            if exponent & 1:
                result = self.multiply(result, value)
            value = self.multiply(value, value)
            exponent >>= 1
        return result

    def trace(self, value):
        result = 0
        for _ in range(self.degree):
            result ^= value
            value = self.multiply(value, value)
        assert result in (0, 1)
        return result

    def polar(self, x, y):
        return self.multiply(self.multiply(x, y), x ^ y)


def check_field_slices(field):
    multiply = field.multiply
    for u, v in product(range(1, field.size), repeat=2):
        first, second = [], []
        for i in range(field.degree):
            c = 1 << i
            first.append((multiply(c, v), multiply(c, u)))
            second.append((
                field.power(multiply(c, v), field.size // 2),
                multiply(c, multiply(u, u)),
            ))
        rows = first + second
        assert all(field.trace(multiply(x, u) ^ multiply(y, v)) == 0 for x, y in rows)
        for vectors, expected in (
            (first, field.degree), (second, field.degree),
```

```
            (rows, 2 * field.degree - 1),
        ):
            assert len(row_basis(x | (y << field.degree) for x, y in vectors)) == expected

    assert len(row_basis(
        multiply(x, y) | (multiply(multiply(x, x), y) << field.degree)
        for x, y in product(range(field.size), repeat=2)
    )) == 2 * field.degree

    for x in range(1, field.size):
        assert [y for y in range(field.size) if not field.polar(x, y)] == [0, x]
    for alpha in range(1, field.size):
        for x in range(field.size):
            radical = all(
                not field.trace(multiply(alpha, field.polar(x, 1 << i)))
                for i in range(field.degree)
            )
            assert radical == (not x or multiply(alpha, field.power(x, 3)) == 1)


def derivative(poly, shift):
    result = set()
    for monomial in poly:
        available = monomial & shift
        deleted = available
        while deleted:
            result.symmetric_difference_update((monomial ^ deleted,))
            deleted = (deleted - 1) & available
    return result


def evaluate(poly, value):
    return sum(term & value == term for term in poly) % 2


def gradient(poly, width):
    result = [set() for _ in range(width)]
    for term in poly:
        for index in bits(term):
            result[index].add(term ^ (1 << index))
    return result


def scalar_gradient(grad, row):
    result = set()
    for index in bits(row):
        result.symmetric_difference_update(grad[index])
    return result


def check_scalar_comparisons(width):
    count = 1 << width
    equations = []
    for a, x in product(range(count), repeat=2):
        row = 0
        for index in bits(a):
            for point in (x ^ a, x, a, 0):
                row ^= 1 << (index * count + point)
        equations.append(row)
    cubic_count = len(list(combinations(range(width), 3)))
    assert width * count - len(row_basis(equations)) == width + width**2 + cubic_count


def check_geometry(width):
```

```
    count = 1 << width
    geometry = []
    for u, v in product(range(count), repeat=2):
        if dot(u, v):
            geometry.append(u | (v << width) | (1 << (2 * width)))
        for x, y, w in product(range(count), range(count), range(2)):
            zero = (y ^ (v if w else 0), x ^ (u if w else 0), dot(u, y) ^ dot(v, x)) == (0,
            ↪  0, 0)
            assert zero == ((x, y, w) in ((0, 0, 0), (u, v, 1)))
    assert len(row_basis(geometry)) == 2 * width + 1


def check_first_profile_normalization(width):
    length = 2 * width + 1
    local_mask = (1 << width) - 1
    last = 1 << (length - 1)

    def polar(a, b):
        u, v = a & local_mask, (a >> width) & local_mask
        x, y = b & local_mask, (b >> width) & local_mask
        first = (y if a & last else 0) ^ (v if b & last else 0)
        second = (x if a & last else 0) ^ (u if b & last else 0)
        return first | (second << width) | (last * (dot(u, y) ^ dot(v, x)))

    coefficients = list(combinations(range(length), 2))

    def row_for(direction, readout):
        return mask_of(
            column for column, (i, j) in enumerate(coefficients)
            if (((direction >> i) & (readout >> j)) ^
                ((direction >> j) & (readout >> i))) & 1
        )

    imposed = []
    for j, i in product(range(length), repeat=2):
        if all(not dot(1 << i, polar(1 << j, 1 << b)) for b in range(length)):
            imposed.append(row_for(1 << j, 1 << i))
    for i, j in combinations(range(width), 2):
        direction = (1 << i) | (1 << j)
        readout = (1 << (width + i)) | (1 << (width + j))
        assert all(not dot(readout, polar(direction, 1 << b)) for b in range(length))
        imposed.append(row_for(direction, readout))
    gauges = [
        mask_of(
            column for column, (i, j) in enumerate(coefficients)
            if dot(1 << i, polar(1 << c, 1 << j))
        )
        for c in range(length)
    ]
    assert len(coefficients) - len(row_basis(imposed)) == length
    assert len(row_basis(gauges)) == length
    assert all(not dot(gauge, row) for gauge, row in product(gauges, imposed))


def check_second_profile(field):
    d = field.degree
    cubic = set()
    for i, j in combinations(range(d), 2):
        output = field.polar(1 << i, 1 << j)
        cubic.update(mask_of((i, j, d + k)) for k in bits(output))
    grad = gradient(cubic, 3 * d)

    def value(point):
        return mask_of(i for i, poly in enumerate(grad) if evaluate(poly, point))
```

```
    for j in range(field.size):
        cube_quadratic = field.power(j, 3)
        for i in bits(j):
            cube_quadratic ^= field.power(1 << i, 3)
        for b in range(field.size):
            n = cube_quadratic ^ field.polar(j, b)
            a = j | (n << (2 * d))
            profile_at_a = value(a) ^ (n << d)
            polar_at_pair = value(a ^ b) ^ value(a) ^ value(b)
            assert profile_at_a == polar_at_pair


def check_triple_cancellation():
    width = 8
    triples = [mask_of(row) for row in combinations(range(width), 3)]
    for degree in (2, 3, 4):
        for indices in combinations(range(width), degree):
            monomial = mask_of(indices)
            grad = gradient({monomial}, width)
            for t in triples:
                observed = derivative(scalar_gradient(grad, t), t)
                expected = {monomial ^ t} if monomial & t == t else set()
                assert observed == expected


def data_instance(field):
    width = field.degree
    cores = [list(range(i * width, (i + 1) * width)) for i in range(4)]
    groups = [block.copy() for block in cores]
    quartic = set()
    next_coordinate = 4 * width
    for k, l in combinations(range(4), 2):
        selector_group, field_group = [j for j in range(4) if j not in (k, l)]
        selector = next_coordinate
        groups[selector_group].append(selector)
        next_coordinate += 1
        for mode in range(2):
            copy = list(range(next_coordinate, next_coordinate + width))
            next_coordinate += width
            groups[field_group].extend(copy)
            for x, y, c in product(range(width), repeat=3):
                first = field.power(1 << x, 1 << mode)
                if field.trace(field.multiply(field.multiply(first, 1 << y), 1 << c)):
                    quartic.add(mask_of((selector, copy[c], cores[k][x], cores[l][y])))
    return cores, groups, quartic, next_coordinate


def check_data(field):
    cores, groups, fixed_quartic, width = data_instance(field)
    core_mask = mask_of(index for block in cores for index in block)
    aux_mask = ((1 << width) - 1) ^ core_mask
    polarity = mask_of(block[-1] for block in cores[:3])
    fixed = fixed_quartic | {polarity}
    real = [block[:-1] for block in cores]
    free = {
        mask_of(indices)
        for degree in (2, 3, 4)
        for chosen in combinations(real, degree)
        for indices in product(*chosen)
    }
    # A linear identity is tested on every free coefficient basis vector
    # and separately on the fixed contribution.
    for coordinate in (0, 1, 2):
```

```
        variable = mask_of(block[coordinate] for block in cores)
        for negative in (0, 1):
            j = variable ^ (polarity if negative else 0)
            beta = j & ~mask_of(groups[0])
            for poly in ({term} for term in free | fixed):
                scalar = scalar_gradient(gradient(poly, width), j)
                literal = evaluate(scalar, j) ^ evaluate(derivative(derivative(scalar, j),
                ↪  beta), 0)
                expected = sum(
                    term.bit_count() >= 3 and term & j == term for term in poly
                ) % 2
                assert literal == expected

            shifted = [derivative(row, j) for row in gradient(fixed, width)]
            contraction = {}
            for index, row in enumerate(shifted):
                for term in row:
                    if term.bit_count() == 2:
                        contraction[term] = contraction.get(term, 0) ^ (1 << index)
            core_rows = [row & core_mask for row in contraction.values()]
            aux_rows = [row & aux_mask for row in contraction.values()]
            assert len(row_basis(core_rows)) == core_mask.bit_count() - 1
            assert all(not dot(row, j) for row in core_rows)
            assert len(row_basis(contraction.values())) == len(row_basis(core_rows)) +
            ↪  len(row_basis(aux_rows))
            for row in annihilator(aux_rows, aux_mask):
                derivative_row = scalar_gradient(shifted, row)
                assert all(term.bit_count() == 1 for term in derivative_row)
                assert not evaluate(derivative_row, j)
                assert not evaluate(derivative_row, beta)

    # A triple guard separates its cubic constant from its quartic
    # linear contraction on each coefficient basis vector.
    triples = set()
    for term in fixed | free:
        triples.update(mask_of(indices) for indices in combinations(list(bits(term)), 3))
    for t in sorted(triples):
        for term in fixed | free:
            if term & t != t:
                continue
            observed = derivative(scalar_gradient(gradient({term}, width), t), t)
            assert observed == {term ^ t}


def main():
    for field in (Field(4, 0b10011), Field(5, 0b100101)):
        check_field_slices(field)
        print(f"GF(2^{field.degree}): trace slices, span and polar radicals checked")
    for width in (3, 4, 5):
        check_scalar_comparisons(width)
    check_geometry(3)
    check_first_profile_normalization(3)
    check_second_profile(Field(4, 0b10011))
    check_triple_cancellation()
    print("comparison ranks, geometry shifts and both auxiliary profiles checked")
    check_data(Field(4, 0b10011))
    print("quartic clause coefficients, triple guards and auxiliary rows checked")


if __name__ == "__main__":
    main()
```

References

[Bab15] László Babai. Graph Isomorphism in Quasipolynomial Time, 2015.

[Car26] Élie Cartan. Sur une classe remarquable d'espaces de Riemann. I. *Bulletin de la Société Mathématique de France*, 54:214–264, 1926. https://doi.org/10.24033/bsmf.1105.

[Car27] Élie Cartan. Sur une classe remarquable d'espaces de Riemann. II. *Bulletin de la Société Mathématique de France*, 55:114–134, 1927. https://doi.org/10.24033/bsmf.1113.

[CS25] Milan Cvrček and David Stanovský. Primitive quandles with alternating displacement group. *Hiroshima Mathematical Journal*, 55:349–359, 2025. https://doi.org/10.32917/h2024010.

[Doy76] P. G. Doyle. On Transitive Graphs. Senior Thesis, April 1976.

[EKR61] Paul Erdős, Chao Ko, and Richard Rado. Intersection theorems for systems of finite sets. *The Quarterly Journal of Mathematics*, 12:313–320, 1961. https://doi.org/10.1093/qmath/12.1.313.

[GBD+26] MIT Hardness Group, Josh Brunner, Erik D. Demaine, Fabian Frei, Timothy Gomez, Markus Hecher, Jeffery Li, and Jayson Lynch. Eth is dead, long live sh, 2026.

[GR01] Chris Godsil and Gordon Royle. *Algebraic Graph Theory*. Springer New York, 2001.

[Hel01] Sigurdur Helgason. *Differential Geometry, Lie Groups, and Symmetric Spaces*. American Mathematical Society, Providence, RI, 2001.

[HEO05] D.F. Holt, B. Eick, and E.A. O'Brien. *Handbook of Computational Group Theory*. Discrete Mathematics and Its Applications. CRC Press, 2005.

[HM67] A. J. W. Hilton and E. C. Milner. Some intersection theorems for systems of finite sets. *The Quarterly Journal of Mathematics*, 18(1):369–384, 1967. https://doi.org/10.1093/qmath/18.1.369.

[Hol81] D. F. Holt. A graph which is edge transitive but not arc transitive. *Journal of Graph Theory*, 5(2):201–204, 1981.

[HSV16] Alexander Hulpke, David Stanovský, and Petr Vojtěchovský. Connected quandles and transitive groups. *Journal of Pure and Applied Algebra*, 220(2):735–758, 2016. https://doi.org/10.1016/j.jpaa.2015.07.014.

[IT16] Yoshitaka Ishihara and Hiroshi Tamaru. Flat connected finite quandles. *Proceedings of the American Mathematical Society*, 144:4959–4971, 2016. https://doi.org/10.1090/proc/13095.

[Jon14] Gareth A. Jones. Primitive permutation groups containing a cycle. *Bulletin of the Australian Mathematical Society*, 89(1):159–165, 2014. https://arxiv.org/abs/1209.5169.

[Kar72] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972. https://doi.org/10.1007/978-1-4684-2001-2_9.

[KNN76] Mikio Kano, Hirosi Nagao, and Nobuo Nobusawa. On finite homogeneous symmetric sets. *Osaka Journal of Mathematics*, 13(2):399–406, 1976. https://ir.library.osaka-u.ac.jp/repo/ouka/all/11786/ojm13_02_11.pdf.

[KS25] Dilpreet Kaur and Pushpendra Singh. Classification of primitive quandles of small order. *Journal of Symbolic Computation*, 131:102452, 2025. https://doi.org/10.1016/j.jsc.2025.102452.

[Loo67] Ottmar Loos. Reflexion spaces and homogeneous symmetric spaces. *Bulletin of the American Mathematical Society*, 73(2):250–253, 1967. https://doi.org/10.1090/S0002-9904-1967-11704-X.

[Loo69] Ottmar Loos. *Symmetric Spaces. I: General Theory*. W. A. Benjamin, 1969. https://www2.math.upenn.edu/~wziller/math661/LoosI.pdf.

[Lub81] Anna Lubiw. Some np-complete problems similar to graph isomorphism. *SIAM Journal on Computing*, 10(1):11–21, 1981.

[Luk93] Eugene M. Luks. Permutation groups and polynomial-time computation. In *Groups and Computation*, volume 11 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 139–175. American Mathematical Society, 1993.

[SB25] Filippo Spaggiari and Marco Bonatto. On core quandles. *Algebra Universalis*, 86:28, 2025. https://doi.org/10.1007/s00012-025-00901-y.

[Ta26] Luc Ta. Graph quandles: Generalized Cayley graphs of racks and right quasigroups. *JoNAS - Journal of Non-Associative Structures*, 1(1):Article 3, 2026. https://doi.org/10.46298/jonas.17215.

[Tak43] Mituhisa Takasaki. Abstraction of symmetric transformations. *Tohoku Mathematical Journal*, 49:145–207, 1943. https://www.jstage.jst.go.jp/article/tmj1911/49/0/49_0_145/_article.

[ZKT85] V. N. Zemlyachenko, N. M. Korneenko, and R. I. Tyshkevich. Graph isomorphism problem. *Journal of Soviet Mathematics*, 29(4):1426–1481, May 1985.