Parallel Complexity of Identifying Groups and Quasigroups via Decompositions

Dan Johnson (College of Charleston)
Michael Levet* (College of Charleston)
Petr Vojtěchovský (University of Denver)
Brett Widholm (College of Charleston)

Relational and Algebraic Methods in Computer Science

April 7, 2026

Graph Isomorphism

Complexity of Graph Isomorphism

- Conjectured to be NP-intermediate (in NP, but neither in P nor NP-complete)
- Algorithmically: $n^{\Theta(\log^2 n)}$-runtime (Babai, 2016).
- Graph Isomorphism (GI) is sandwiched between linear algebra and multilinear algebra:
  - Lower bound: DET ($\mathsf{NL} \subseteq \mathsf{DET} \subseteq \mathsf{TC}^1$).
  - Upper-bound: $\mathbb{F}$-Tensor Isomorphism ($\mathsf{TI}_{\mathbb{F}}$).
  - When $\mathbb{F}$ is finite, $\mathsf{TI}_{\mathbb{F}} \subseteq \mathsf{NP} \cap \mathsf{coAM}$.

Quasigroup Isomorphism

Background

- Best algorithmic bound:
  - Quasigroup Isomorphism (QGpI): $n^{\log(n)+O(1)}$-time [Mil78].
  - Group Isomorphism (GpI): $n^{(1/4)\log(n)+O(1)}$-time [Ros13, Luk15].
- Best complexity-theoretic bound (for QGpI and GpI):

$$\exists^{\log^2 n}\forall^{\log n}\exists^{\log n}\mathsf{DTISP}(\mathrm{polylog}(n), \log(n)),$$

which can be simulated by depth-4 AC circuits of size $n^{O(\log n)}$ [CG**L**W, 2025].

Quasigroup Isomorphism

Background

- Lower-bounds against depth-2 AC circuits remain open.
- QGPI is strictly easier than GI under AC^0-reductions [CTW13].
- Long history of NC algorithms for GI, but nascent for groups and quasigroups.
- Considerable work on polynomial-time isomorphism tests for important families of groups, but essentially no work on special cases of QGPI (beyond $O(1)$-generated quasigroups).

Graph Isomorphism

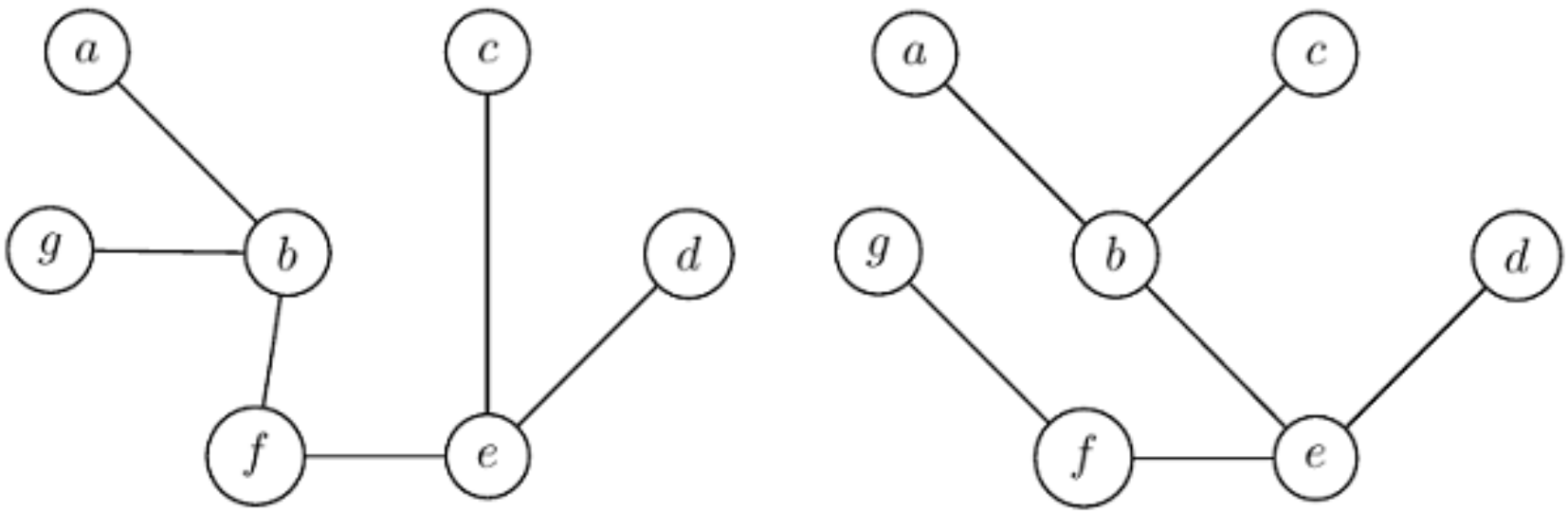
Techniques

How do we test graphs for isomorphism algorithmically?

- Color-Refinement (Weisfeiler–Leman)
- Permutation Group Algorithms

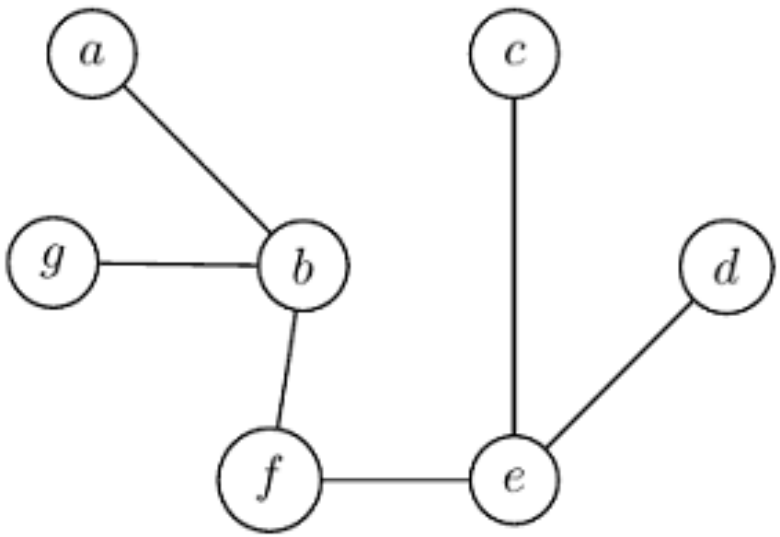
1-Dimensional Weisfeiler–Leman: Example

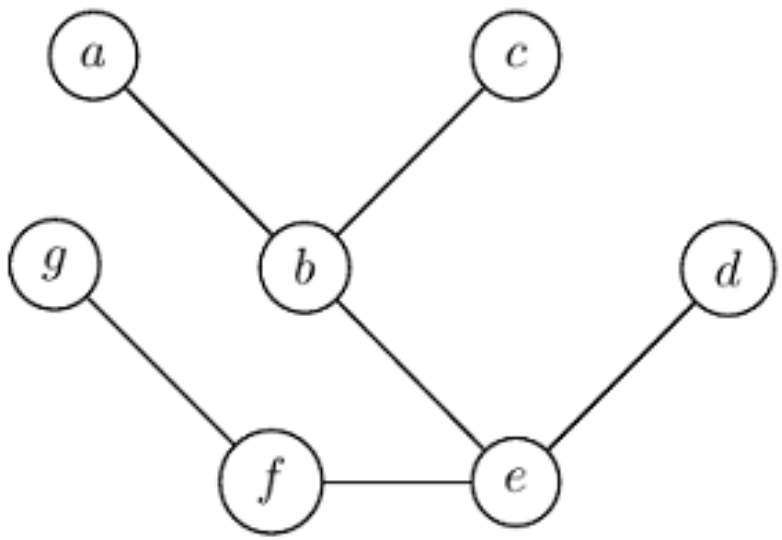
Input:



1-Dimensional Weisfeiler–Leman: Example

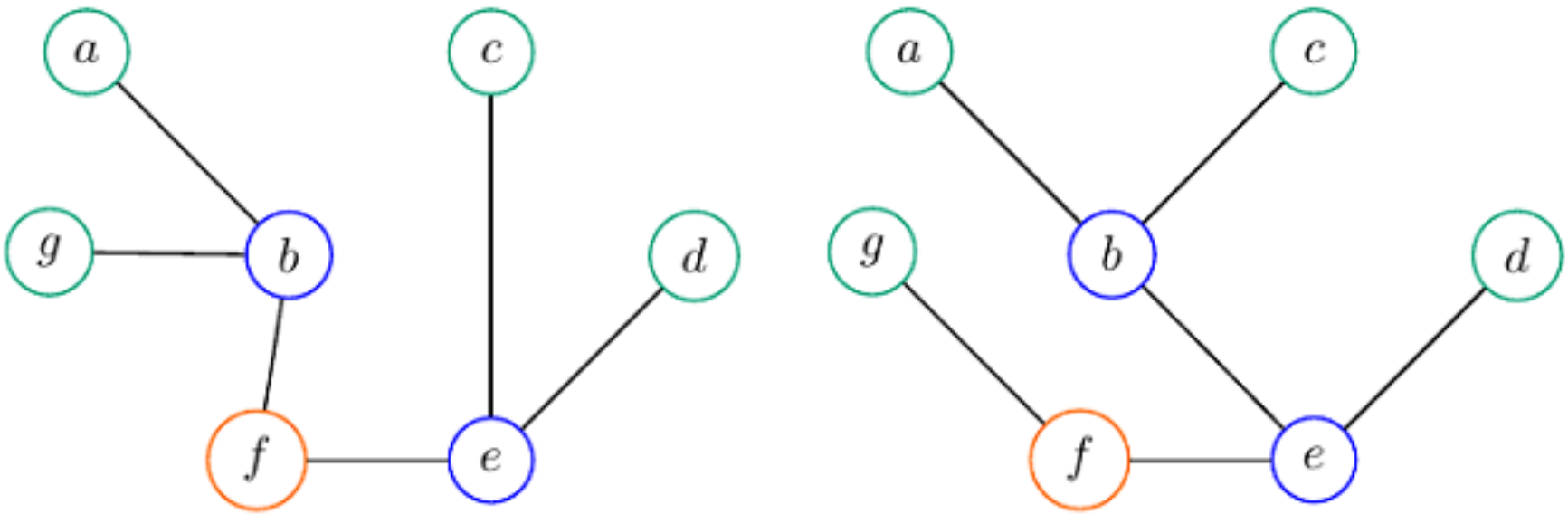
Initial Coloring: Use coloring provided on the graph





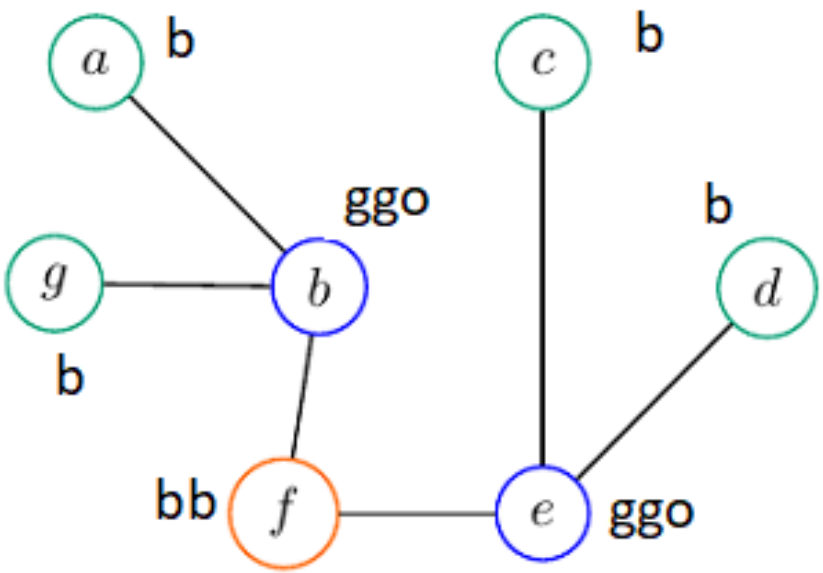
1-Dimensional Weisfeiler–Leman: Example

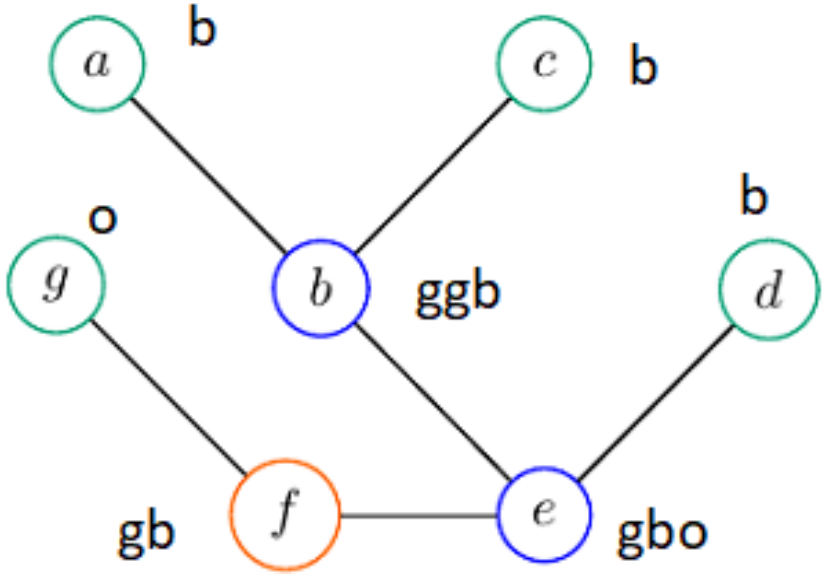
Refinement Step: Color each vertex based on its initial color and multiset of colors of its neighbors.



Weisfeiler–Leman: Example

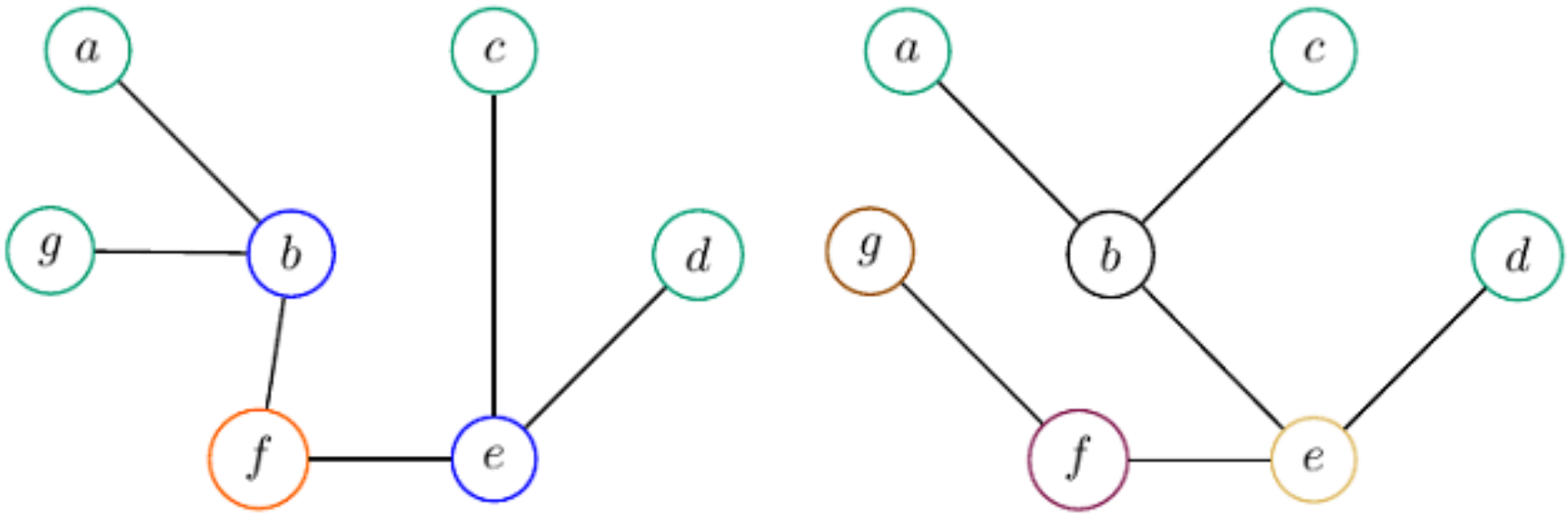
Refinement Step: Color each vertex based on its initial color and multiset of colors of its neighbors.





Weisfeiler–Leman: Example

Refinement Step: Color each vertex based on its initial color and multiset of colors of its neighbors.



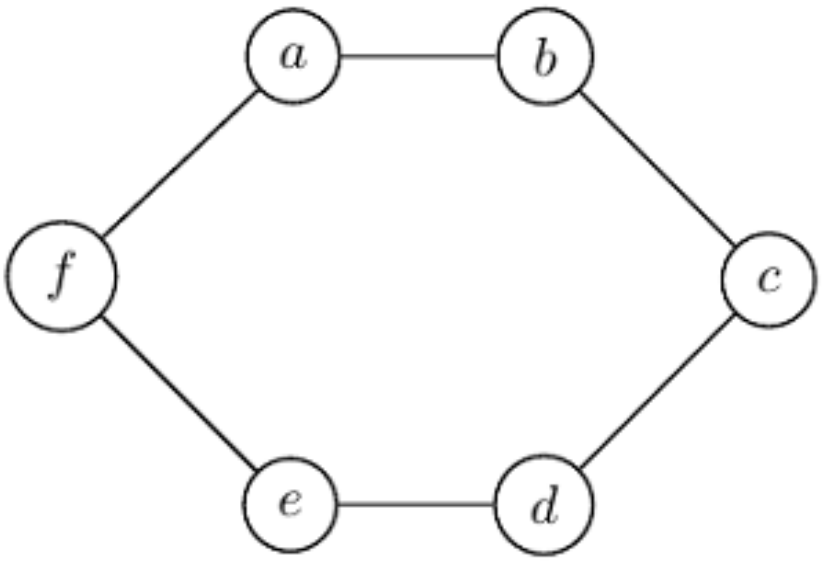
Weisfeiler–Leman

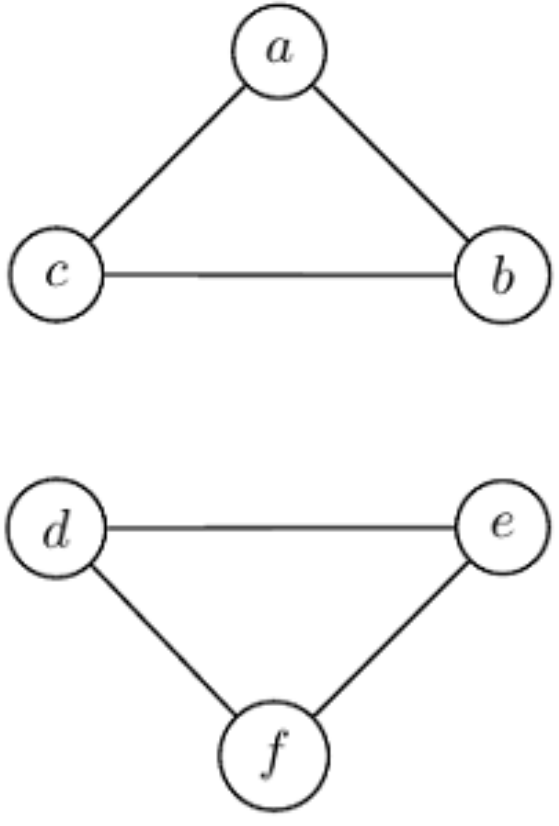
1-Dimensional Weisfeiler–Leman

- Initial Coloring: Use coloring provided on the graph.
- Refinement Step: Two vertices u, v receive the same color at round $r + 1$ if:
  - They have the same color at round r, and
  - The multiset of colors of u's neighbors is the same as for v's neighbors.
- Termination: If the multiset of colors differ at the end of a given round, we conclude that the two graphs are non-isomorphic.

Weisfeiler–Leman: Example

Counter-Example: 1-Dimensional Weisfeiler–Leman fails to distinguish C_6 from $2K_3$. All vertices have degree 2 and so receive the same initial color.





Weisfeiler–Leman

Higher-Dimensional Weisfeiler–Leman

Fix $k \geq 2$. We consider the k-dimensional Weisfeiler–Leman (k-WL). We color all k-tuples of vertices.

- Initial Coloring: Marked isomorphism type of induced subgraph.
- Refinement Step: The color of $\overline{u}$ at round $r+1$ depends on:
  - Color of $\overline{u}$ at round r.
  - The multiset of colors of *nearby* (Hamming distance 1) k-tuples, where colors are from round r.

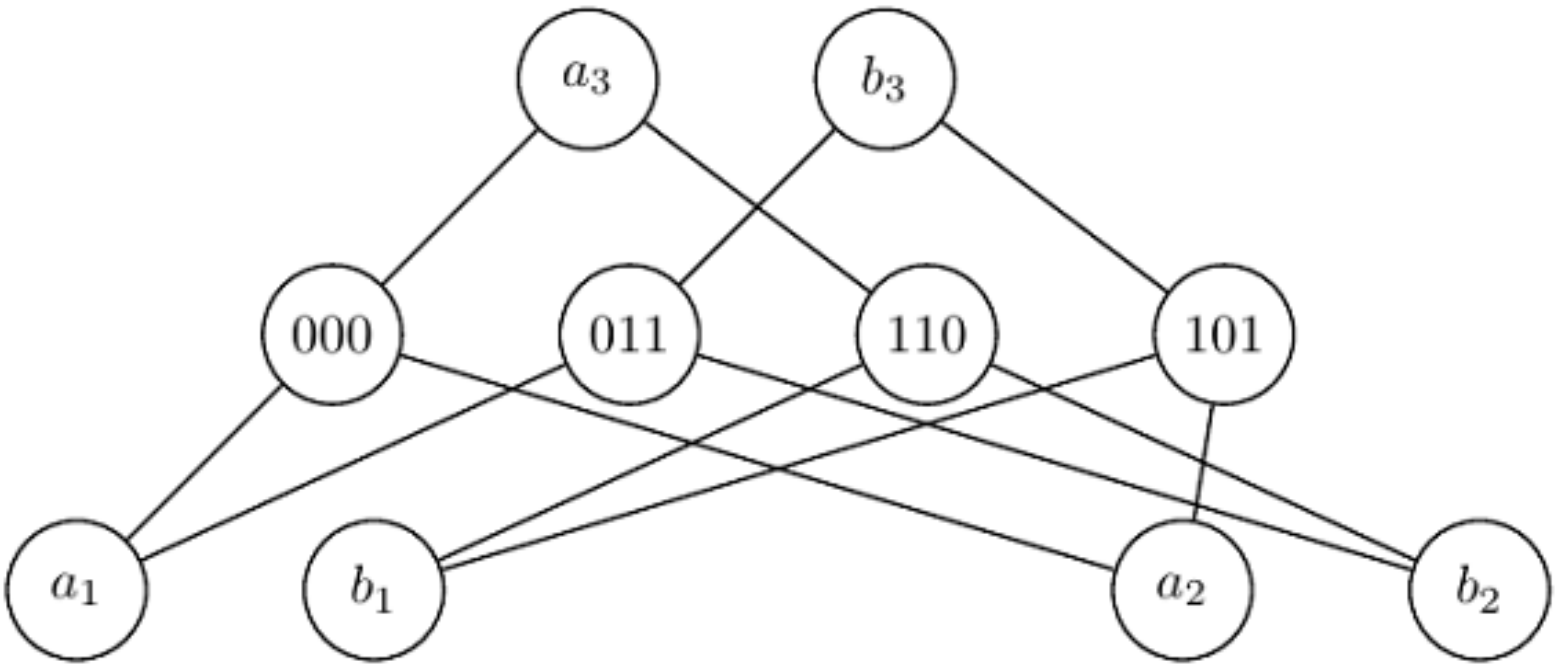
Weisfeiler–Leman

State of the Art

- 1-Dimensional Weisfeiler–Leman (1-WL) is used in most practical algorithms (e.g., `nauty`, `traces`, `saucy`)
- 1-WL identifies almost all graphs
- Babai's $n^{\Theta(\log^2 n)}$-runtime graph isomorphism test combines higher-dimensional Weisfeiler–Leman with techniques from group theory (2016).
- Higher-dimensional Weisfeiler–Leman is unable to efficiently handle Graph Isomorphism in general.

Weisfeiler–Leman

Counter-Example: Higher-dimensional Weisfeiler–Leman gets stuck on a construction of Cai, Fürer, & Immerman (1992). Replace each vertex with a *gadget* and connect gadgets based on edges of original graph.



Weisfeiler–Leman

Characterizations

Weisfeiler–Leman is equivalent to the following:

- 1-ary Ehrenfeucht–Fraïssé Bijective Pebble Game.
- First-Order Logic with Counting Quantifiers (FO + C)
- Sherali–Adams and Lasserre Hierarchies (Linear Programming)
- (Generalized) Coherent Configurations
- Restriction of Polynomial Calculus (Algebraic Proof Complexity)
- Graph Neural Networks

Weisfeiler–Leman for Groups

WL for Groups

Brachter and Schweitzer [BS20] introduced three variants of WL for groups. We will focus on:

- **WL Version I:** Initial coloring is AC^0-computable.
  - $(g_1, \ldots, g_k)$ and $(h_1, \ldots, h_k)$ receive the same initial color iff: (a) $g_i = g_j \iff h_i = h_j$, and (b) $g_i g_j = g_m \iff h_i h_j = h_m$, for all $i, j, m \in [k]$.
- **WL Version II:** Initial coloring is logspace-computable.
  - $(g_1, \ldots, g_k)$ and $(h_1, \ldots, h_k)$ receive the same initial color iff the map $g_i \mapsto h_i$ (for all $i \in [k]$) extends to an isomorphism of $\langle g_1, \ldots, g_k \rangle$ and $\langle h_1, \ldots, h_k \rangle$.
- For both WL Versions I and II, the refinement step is identical for graphs.

Weisfeiler–Leman and Direct Products

Theorem (Brachter–Schweitzer, ESA 2022)

Let $G = G_1 \times \cdots \times G_d$ be a decomposition into indecomposable direct factors, and let $k \geq 5$. If G and H are not distinguished by k-WL Version II, then there exist direct factors $H_i \leq H$ such that $H = H_1 \times \cdots \times H_d$ such that for all $i \in [d]$, G_i and H_i are not distinguished by $(k-1)$-WL Version II.

Theorem (Grochow–L.; FCT'23, *J. Comput. Syst. Sci.* 2026)

Let $G = G_1 \times \cdots \times G_d$ be a decomposition into indecomposable direct factors, let $k \geq 5$, and let $r := r(n)$. If G and H are not distinguished by $(k, r + O(\log n))$-WL Version II, then there exist direct factors $H_i \leq H$ such that $H = H_1 \times \cdots \times H_d$ such that for all $i \in [d]$, G_i and H_i are not distinguished by $(k-1, r)$-WL Version II.

Weisfeiler–Leman and Direct Products

Theorem (Grochow–L.; FCT'23, *J. Comput. Syst. Sci.* 2026)

There exists an infinite family $(G_m, H_m)_{n\geq 1}$, where $G_m \ncong H_m$ are Abelian groups are of the same order, and count-free WL requires dimension at least $(1/3)\log_2 |G_m|$ to distinguish G_m from H_m.

Weisfeiler–Leman and Direct Products

Question

What is the power of count-free WL to handle direct products?

Weisfeiler–Leman and Direct Products

Theorem (Collins–L., *Int. J. Algebra Comput.* 2024)

There exists a uniform $\forall^{\log n} MAC^0(FOLL)$ algorithm to decide isomorphism between: a group G that decomposes as a direct product of non-Abelian simple groups, and an arbitrary group H.

Proof.

- Run $O(1)$-dimensional count-free WL for $O(\log \log n)$ rounds (FOLL).
- Use $O(\log n)$ universally quantified co-nondeterministic bits ($\forall^{\log n}$) to specify a color class C (intuition: specify generators for a direct factor of G).
- Use a single Majority gate to compare the number of tuples from G vs. H belonging to C (MAC^0).

Weisfeiler–Leman and Direct Products

Theorem (Johnson–L.–Vojtěchovský–Widholm, 2026)

The $O(1)$-dimensional count-free WL Version I identifies, in $O(\log \log n)$ rounds, all groups decomposing as a direct product of non-Abelian simple groups.

Corollary (Johnson–L.–Vojtěchovský–Widholm, 2026)

There exists a uniform FOLL algorithm to decide isomorphism between: a group G that decomposes as a direct product of non-Abelian simple groups, and an arbitrary group H.

Weisfeiler–Leman and Direct Products

Definition

Let $\mathcal{C}$ be the class of groups $G = G_1 \times \cdots \times G_d$, where each G_i ($i \in [d]$) is directly indecomposable and:

- $O(1)$-generated, and
- Perfect ($G_i = [G_i, G_i]$) or Centerless ($Z(G_i) = \{1\}$).

Note that the above direct product decomposition of G is unique (up to reordering of the factors).

Weisfeiler–Leman and Direct Products

Theorem (Johnson–L.–Vojtěchovský–Widholm, 2026)

The following serve as complete isomorphism tests for $\mathcal{C}$:

- *The $O(1)$-dimesional count-free WL Version II, run for $O(\log \log n)$ rounds.*
- *The $O(1)$-dimensional counting WL Version II, run for $O(1)$ rounds.*

Weisfeiler–Leman and Direct Products

Theorem (Grochow–L.; FCT'23, *J. Comput. Syst. Sci.* 2026)

There exists a uniform TC^1 algorithm to decide isomorphism between: a group $G \in \mathcal{C}$ and an arbitrary group H.

Corollary (Johnson–L.–Vojtěchovský–Widholm, 2026)

There exists a logspace algorithm to decide isomorphism between: a group $G \in \mathcal{C}$ and an arbitrary group H.

Central Quasigroups

Definition (Central Quasigroups)

A quasigroup $(Q, *)$ is *central* if there is an Abelian group $(Q, +)$, automorphisms ϕ, ψ of $(Q, +)$, and $c \in (Q, +)$ such that:

$$x * y = \phi(x) + \psi(y) + c,$$

for all $x, y \in Q$. We denote the corresponding quasigroup by $\mathcal{Q}(Q, +, \phi, \psi, c)$.

Central Quasigroups

Theorem (Johnson–L.–Vojtěchovský–Widholm, 2026)

There exists a uniform NC algorithm to decide isomorphism between a central quasigroup G and an arbitrary quasigroup H.

Central Quasigroups

Theorem (Stanovský–Vojtěchovský, 2015)

Let $(G,+)$ be an Abelian group, let $\phi_1,\psi_1,\phi_2,\psi_2 \in Aut(G,+)$, and let $c_1,c_2 \in G$. Then the following statements are equivalent:

1. *the central quasigroups $\mathcal{Q}(G,+,\phi_1,\psi_1,c_1)$ and $\mathcal{Q}(G,+,\phi_2,\psi_2,c_2)$ are isomorphic,*
2. *there is an automorphism $\gamma \in Aut(G,+)$ and an element $u \in Im(1-\phi_1-\psi_1)$ such that*

$$\phi_2 = \gamma\phi_1\gamma^{-1},\ \psi_2 = \gamma\psi_1\gamma^{-1},\ c_2 = \gamma(c_1+u).$$

Central Quasigroups

Proof (Sketch).

- Decide if quasigroups G, H are central, and compute decompositions $\mathcal{Q}(G, +, \phi_1, \psi_1, c_1), \mathcal{Q}(H, +, \phi_2, \psi_2, c_2)$ if they exist.
- Decide if $(G, +) \cong (H, +)$ by computing bases $b = (b_1, \ldots, b_k)$ for $(G, +)$, and $b' = (b'_1, \ldots, b'_k)$ for $(H, +)$.
- Let:

$$\bar{x} = (c_1, \phi_1(b_1), \ldots, \phi_1(b_k), \psi_1(b_1), \ldots, \psi_1(b_k))$$
$$\bar{y} = (c_2, \phi_2(b'_1), \ldots, \phi_2(b'_k), \psi_2(b'_1), \ldots, \psi_2(b'_k)).$$

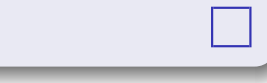
Central Quasigroups

Proof (Sketch).

- For $u \in \mathrm{Im}(1 - \phi_1 - \psi_1)$, let:

$$\bar{x}_u = (c_1 + u, \phi_1(b_1), \ldots, \phi_1(b_k), \psi_1(b_1), \ldots, \psi_1(b_k)).$$

- For each $u \in \mathrm{Im}(1 - \phi_1 - \psi_1)$, use Pointwise Transporter to decide if there exists $\gamma \in \mathrm{Aut}(G, +)$ such that $\bar{x}_u^\gamma = \bar{y}$.
- As we have $2k + 1 \in O(\log |G|)$ points, Pointwise Transporter is NC-computable.



Open Questions

Question

Does isomorphism testing of central quasigroups belong to logspace?

Question

Consider the class of groups, where each group G admits a unique direct product decomposition such that each indecomposable direct factor is also $O(1)$-generated. Does this class have bounded (count-free) WL-dimension?