WEISFEILER–LEMAN FOR EMBEDDED GRAPHS: TOPOLOGICAL MESSAGE PASSING ON SURFACES

Anonymous authors
Paper under double-blind review

ABSTRACT

We study message passing architectures for graphs embedded on surfaces and thereby connect graph machine learning with topological graph theory. While previous Topological Deep Learning methods study the intrinsic topology of graphs/complexes determined by their combinatorics, we make use of the topology of their embeddings and position our approach between the abstract combinatorial and the explicit geometric settings. This is relevant for many application areas such as drug design. On the one hand, the pure connectivity information of molecules is often not sufficient for chemically interesting predictions; on the other hand, the full 3D structure is not needed and can in many cases hinder learning. For graphs on surfaces, we propose three methods to exploit the embedding via its induced rotation system. We prove the following: (i) Sparse cyclic aggregations identify all embedded trees, (ii) Dense, with a ternary relations on the vertices identify all embedded graphs at dimension three and not below, (iii) Binary relations on the oriented edges achieve the same result already at dimension two. Finally, we fully characterize how these methods compare in expressivity. We propose graph neural networks motivated by these results and evaluate them on topological property prediction tasks and chirality-sensitive molecular tasks. Our experiments show that our approach is not only theoretically interesting but also practically relevant.

1 INTRODUCTION

We initiate the study of message-passing architectures for graphs embedded on surfaces. Such embeddings provide information beyond abstract connectivity, ranging from metric geometry to topology. Standard message-passing graph neural networks (GNNs) ignore this information and therefore cannot distinguish between different embeddings of the same graph. Geometric GNNs address this limitation by incorporating spatial information while ensuring invariance or equivariance under translations, rotations, and, depending on the task, reflections (Bronstein et al., 2021; Satorras et al., 2021; Joshi et al., 2023). However, for many tasks, including chirality-sensitive prediction on molecular datasets (Gainski et al., 2023), retaining the full metric geometry is unnecessary, what matters is the topology of the *embedding*. For instance, the graphs and are *topologically* equivalent embeddings, but geometric GNNs may assign them distinct representations. Conversely, and are non-equivalent embeddings of the same abstract graph and hence standard GNNs must assign them exactly the same representation. In our work, we address this topological notion of graph equivalence by studying message passing architectures for graphs *cellularly embedded on oriented surfaces*, thereby connecting graph machine learning with a classic field of mathematics: *topological graph theory*. Leaning on a result by Edmonds Jr (1960), we encode a cellular embedding via its induced *rotation system*, i.e., the cyclic order in which the neighbors of each vertex appear around it on the surface, in order to input the topological information into the GNNs. We propose and characterize the expressivity of three Weisfeiler–Leman (WL) based methods that account for the rotation system in different ways.

We (i) present *Cyclic Weisfeiler–Leman* (CWL), a message-passing algorithm that at each node aggregates information from its neighbors in the cyclic order provided by the rotation system. We show that CWL is complete for embedded trees, but fails on simple non-equivalent embedded planar graphs. To address this expressivity limitation, we run ordinary k-dimensional WL (k-WL) on two relational structures that encode the embedding information. We (ii) introduce k-TWL$^{\mathcal{R}}$, which acts on the graph with an additional ternary relation that encodes whether two vertices immediately

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Summary of our theoretical results. ✓ and × denote whether the algorithm can distinguish the graphs. The symbols $\simeq$, $\not\simeq$, $\preceq$, $\prec$ denote *equally expressive, not comparable, at most as expressive, less expressive*, respectively. See Section 5.3 for more details.

Embedded graphs	$\{1\text{-TWL}^{\mathcal{D}}$	$\not\simeq$	CWL$\}$	$\prec$	2-TWL$^{\mathcal{R}}$	$\prec$	$\{3\text{-TWL}^{\mathcal{R}}$	$\simeq$	2-TWL$^{\mathcal{D}}\}$
All trees	×		✓		✓		✓		✓
A planar pair (Figure 2b)	✓		×		✓		✓		✓
A planar/toroidal pair (Figure 2c)	×		×		×		✓		✓
All graphs	×		×		×		✓		✓

follow each other around a common neighbor. We (iii) introduce k-TWL$^{\mathcal{D}}$, which acts on the darts, i.e., oriented edges, with two binary relations, one encoding reversal and the other whether one dart immediately follows the other around the common tail vertex. We prove that 3-TWL$^{\mathcal{R}}$ and 2-TWL$^{\mathcal{D}}$ are complete on all embedded graphs, that is, they can distinguish all pairs of non-equivalently embedded graphs. Our result sharpens the upper bound of Grohe & Verbitsky (2006), whose results imply the completeness of 4-TWL$^{\mathcal{R}}$, and settles the exact WL dimension of embedded graphs at 3. We complement our expressivity analysis by showing that 2-TWL$^{\mathcal{R}}$ is strictly more expressive than CWL and 1-TWL$^{\mathcal{D}}$, and that the latter two are incomparable. We summarize our theoretical contributions in Table 1.

The rest of the paper is organized as follows. In Section 2 we discuss the related work. In Section 3 we provide the necessary background and preliminaries. Section 4 presents CWL and investigates its expressivity, while Section 5 presents and analyzes k-TWL$^{\mathcal{R}}$ and k-TWL$^{\mathcal{D}}$. We discuss our experiments in Section 6 and draw our conclusions in Section 7.

2 Related work

WL on surface-embeddable graphs. The expressivity of WL has long been studied on graph classes defined by surface embeddability (Grohe, 2000). The WL dimension of abstract planar graphs is at most 3 (Kiefer et al., 2019), while graphs of bounded genus admit a dimension bound linear in their genus (Grohe & Kiefer, 2019). The tractability of graph isomorphism on such graph classes has inspired architectures specialized to abstract planar (Dimitrov et al., 2023) and outerplanar graphs (Bause et al., 2025). In these cases, however, only the abstract graph is given: the embedding is not part of the input and thus non-equivalent embeddings of the same graph cannot be distinguished.

Topological deep learning. Topological deep learning combines methods from Topological Data Analysis and Machine Learning (Papamarkou et al., 2024; Hajij et al., 2022). One line of work augments graph representations using persistent homology (Hofer et al., 2020; Zhao et al., 2020; Horn et al., 2022; Ballester & Rieck, 2024). Other approaches have developed neural architectures for higher-order domains, including simplicial (Ebli et al., 2020; Roddenberry et al., 2021; Bodnar et al., 2021b) and cellular or combinatorial complexes (Bodnar et al., 2021a; Hajij et al., 2022; Papillon et al., 2023). Eitan et al. (2025) show that standard higher-order message passing can fail to distinguish complexes differing in fundamental topological properties, while Akbari et al. (2026) characterize higher-dimensional WL tests on combinatorial complexes through counting logic and pebble games. Although these approaches may exploit the intrinsic topology determined by the abstract combinatorial object, they do not consider it as a geometrical/topological object embedded in an ambient space (e.g., Euclidean space). Therefore, any geometrical/topological notion about its particular embedding, such as equivalence under *homeomorphism of pairs* (see Section 3), is lost. At the other extreme, some architectures incorporate metric coordinates while maintaining Euclidean equivariance (Eijkelboom et al., 2023; Battiloro et al., 2025), thereby retaining metric information beyond the topology of the embedding.

Embedded graphs and ordered message passing. In our work, we focus on graphs cellularly embedded on oriented surfaces. In particular, the concepts of continuity and homeomorphism of pairs, which, informally, are continuous reversible deformations within the ambient space, are central: we do not allow edge crossings nor vertices jumping on the other side of an edge within the surface. The ternary encoding of a rotation system is introduced by Grohe & Verbitsky (2006), but their focus

is the descriptive and parallel complexity of graph isomorphism rather than message passing. In our investigation, we rely on both their approach and on a cyclically ordered aggregation. Different GNNs distinguish neighbors through a fixed local numbering of incident edges for combinatorial optimization and expressivity analysis (Sato et al., 2019; Garg et al., 2020). Spiral convolutions, instead, arrange local neighborhoods into fixed sequences on 3D meshes sharing the same connectivity (Bouritsas et al., 2019). Also closely related, ChiENN (Gainski et al., 2023) introduces a cyclically ordered aggregation derived from the three-dimensional molecular geometry to capture tetrahedral chirality. Unlike ChiENN, our cyclic orders are induced by the cellular embedding.

3 Background

This section introduces the necessary background. In particular, we discuss the k-WL algorithm on relational structures and its connection to the pebble game. We then discuss a classical result of Edmonds, the starting point of our work, which turns topological statements about an embedded graph into combinatorial statements about its rotation system. We consider closed connected oriented[1] surfaces $\mathcal{S}$ and simple finite undirected graphs $G = (V, E)$, with $E \neq \varnothing$, identified with their geometric realizations as 1-dimensional simplicial complexes. We use "$\cong$" both for isomorphism of graphs and for homeomorphism of topological spaces (i.e., for the isomorphism in the respective categories). We assume familiarity with the WL algorithm and the concept of expressivity used to compare algorithms. Both are reviewed in Appendix A.

Relational structures and the k-WL algorithm. A finite relational vocabulary ξ consists of relation symbols R with fixed arities $a(R)$. A finite relational structure with vocabulary ξ is a tuple $\mathcal{A} = \langle V(\mathcal{A}), (R^{\mathcal{A}})_{R\in\xi}\rangle$, where $V(\mathcal{A})$ is a finite set called *universe* or *vertex set* and $R^{\mathcal{A}} \subseteq V(\mathcal{A})^{a(R)}$. In particular, a graph is a relational structure with a single binary relation. An isomorphism between two relational structures with the same relational vocabulary is a bijection between the respective vertex sets that preserves every relation in both directions. For a k-tuple $\bar{v} = (v_1, \dots, v_k) \in V(\mathcal{A})^k$, its *atomic type* is the Boolean vector

$$\mathrm{atp}_{\mathcal{A}}(\bar{v}) \coloneqq \left(\left(\mathbb{1}[v_i = v_j]\right)_{i,j\in[k]}, \left(\mathbb{1}[(v_{i_1}, \dots, v_{i_r}) \in R^{\mathcal{A}}]\right)_{\substack{R\in\xi,\ r=a(R) \\ (i_1,\dots,i_r)\in[k]^r}} \right).$$

Thus, it records all equalities and relation instances among the entries of $\bar{v}$ (indices may repeat). We denote by $\bar{v}[i \leftarrow w]$ the tuple obtained from $\bar{v}$ by replacing v_i with w. The k-dimensional Weisfeiler–Leman algorithm (k-WL)[2] initializes the colors of the k-tuples as $c^{(0)}_{\bar{v}} = \mathrm{atp}_{\mathcal{A}}(\bar{v})$ and uses the following aggregation and update:

$$\mathcal{M}^{(t)}(\bar{v}) = \left\{\!\!\left\{ \left(\mathrm{atp}_{\mathcal{A}}(\bar{v}, w), \left(c^{(t)}_{\bar{v}[i\leftarrow w]}\right)_{i=1}^{k}\right) : w \in V(\mathcal{A}) \right\}\!\!\right\}, \qquad c^{(t+1)}_{\bar{v}} = \textsc{hash}\left(c^{(t)}_{\bar{v}}, \mathcal{M}^{(t)}(\bar{v})\right)$$

where HASH is as defined above. Writing $c_{\bar{v}}$ for the stable color, we define k-WL$(\mathcal{A}) \coloneqq \{\!\{c_{\bar{v}} : \bar{v} \in V(\mathcal{A})^k\}\!\}$ and write $\mathcal{A} \equiv_{k\text{-WL}} \mathcal{B}$ if k-WL$(\mathcal{A}) = k$-WL$(\mathcal{B})$. We say that k-WL distinguishes $\mathcal{A}$ and $\mathcal{B}$ if $\mathcal{A} \not\equiv_{k\text{-WL}} \mathcal{B}$. Specializing to graphs and to $k = 1$, the resulting 1-WL has the same graph-distinguishing power as the WL algorithm above. We also define the *count-free* variant k-WL$_{cf}$, in which the multiset in each refinement aggregate is replaced by its underlying set.

Remark 3.1. The term $\mathrm{atp}_{\mathcal{A}}(\bar{v}, w)$ that records the complete atomic type of the $(k+1)$-tuple $(\bar{v}, w)$ is commonly omitted in related work. Indeed, when every relation has arity at most k, the aggregate containing $\mathrm{atp}_{\mathcal{A}}(\bar{v}, w)$ is determined by the current color and the same aggregate without $\mathrm{atp}_{\mathcal{A}}(\bar{v}, w)$, and therefore it carries no additional information. We keep it to account for relations of arity $k+1$. In particular, the term is necessary to establish the equivalence between WL and 1-WL.

The pebble game. Let $\mathcal{A}$ and $\mathcal{B}$ be relational structures with the same relational vocabulary ξ, and let $k \geq 1$. The k-pebble Ehrenfeucht–Fraïssé game, denoted $\mathrm{EF}^k(\mathcal{A}, \mathcal{B})$, is played by two players, *Spoiler* and *Duplicator*, with pebbles $p_1, \dots, p_k$. Each pebble has two copies, one reserved for $\mathcal{A}$ and one for $\mathcal{B}$. Initially, no pebble is placed. Each round consists of two moves:

1. Spoiler chooses an index $i \in [k]$ and a structure $\mathcal{C} \in \{\mathcal{A}, \mathcal{B}\}$, and places the $\mathcal{C}$-copy of p_i on an element of $V(\mathcal{C})$, repositioning it if necessary.

[1] Green marks the assumptions adopted throughout the paper.

[2] In the literature, this version of the algorithm is also known as Folklore k-WL.

2. Duplicator places the other copy of p_i on an element of the other structure, likewise moving it if necessary.

Distinct pebbles may occupy the same element. If p_i has been placed, let $x_i \in V(\mathcal{A})$ and $y_i \in V(\mathcal{B})$ denote the positions of its two copies. Duplicator has a *winning strategy* in $\mathrm{EF}^k(\mathcal{A}, \mathcal{B})$ if Duplicator can ensure that, after every round, the correspondence $\{(x_i, y_i) : p_i \text{ is placed}\}$ is a partial isomorphism between $\mathcal{A}$ and $\mathcal{B}$. Explicitly, for all placed indices i, j, $x_i = x_j \iff y_i = y_j$, and, for every $R \in \xi$ and every $(i_1, \dots, i_{a(R)})$ tuple of placed indices, we have $(x_{i_1}, \dots, x_{i_{a(R)}}) \in R^{\mathcal{A}} \iff (y_{i_1}, \dots, y_{i_{a(R)}}) \in R^{\mathcal{B}}$, where indices may repeat. Spoiler has a *winning strategy* if Spoiler can force a violation of these conditions after finitely many rounds.

The following result establishes the connection between pebble games and WL.

Theorem 3.2 (Immerman 1999; Cai et al. 1992)**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be two relational structures with the same vocabulary and let $k \geq 1$. If Spoiler has a winning strategy in $\mathrm{EF}^{k+1}(\mathcal{A}, \mathcal{B})$, then (count-free) k-WL distinguishes $\mathcal{A}$ and $\mathcal{B}$.*

Embeddings. An *embedding* of G on $\mathcal{S}$ is a continuous injective map $\Gamma\colon G \hookrightarrow \mathcal{S}$ that is a homeomorphism onto its image. We write (G, Γ) for an embedded graph $\Gamma(G) \subset \mathcal{S}$. An embedding is *cellular* if every connected component of $\mathcal{S} \setminus \Gamma(G)$ is homeomorphic to an open disk. These components are called *faces*. We work exclusively with cellular embeddings, as they allow us to recover the surface just from the local embedding information[3]. As a consequence of cellularity, all our graphs are connected. Two graphs G and G' embedded on $\mathcal{S}$ and $\mathcal{S}'$ via Γ and Γ', respectively, are *homeomorphic as pairs*, written $(G, \Gamma) \cong_H (G', \Gamma')$, if there exists an abstract isomorphism $\varphi\colon G \to G'$ and a homeomorphism $h\colon \mathcal{S} \to \mathcal{S}'$ such that $h \circ \Gamma = \Gamma' \circ \varphi$. A homeomorphism of pairs is *orientation-preserving*, written $(G, \Gamma) \cong_{H_+} (G', \Gamma')$, if the ambient homeomorphism h preserves the orientations of $\mathcal{S}$ and $\mathcal{S}'$.

Rotation Systems. A *rotation system* for a simple graph $G = (V, E)$ is a family $\pi = (\pi_v)_{v \in V}$, where each π_v specifies a cyclic order on the neighborhood $N(v)$. We also view π_v as the cyclic permutation mapping each neighbor to its successor in this order, and write (G, π) for the graph equipped with its rotation system. (G, π) and (G', π') are *equivalent*, written $(G, \pi) \cong_R (G', \pi')$, if there exists a graph isomorphism $\varphi\colon G \to G'$ such that $\pi'_{\varphi(v)}\big(\varphi(u)\big) = \varphi\big(\pi_v(u)\big)$ for every $v \in V(G)$ and $u \in N(v)$ (we say that φ *commutes* with π and π'). Given an embedding (G, Γ) on a closed oriented surface $\mathcal{S}$, the orientation of $\mathcal{S}$ induces a cyclic order on the neighbors of each vertex, and hence an *induced rotation system* π. The following theorem allows us to use rotation systems to fully capture cellular embeddings up to orientation-preserving homeomorphisms of pairs.

Theorem 3.3 (Edmonds Jr 1960)**.** *Let $\Gamma\colon G \hookrightarrow \mathcal{S}$ and $\Gamma'\colon G' \hookrightarrow \mathcal{S}'$ be cellular embeddings on closed connected oriented surfaces $\mathcal{S}$ and $\mathcal{S}'$, and let π and π' be their induced rotation systems. Then $(G, \pi) \cong_R (G', \pi')$ if and only if $(G, \Gamma) \cong_{H_+} (G', \Gamma')$.*

It follows that π determines the ambient surface up to homeomorphism, allowing us to represent an embedded graph by (G, π), omitting $\mathcal{S}$. We use two additional representations of (G, π).

- The *rotation structure* is a relational structure $\mathcal{R} := \langle V, E, T \rangle$ (or simply $\langle G, T \rangle$), where $E(x, y) \iff xy \in E(G)$ and $T(v, u, w) \iff \pi_v(u) = w$.
- Let $D = \{(u, v) : uv \in E(G)\}$ be the set of oriented edges, called *darts*, where u and v are the *tail* and the *head*, respectively. Define the permutations $\sigma, \tau\colon D \to D$ by $\sigma(u, v) = (u, \pi_u(v))$ and $\tau(u, v) = (v, u)$: they map each dart to the cyclic successor around the tail and to its reverse, respectively. The *dart structure* is a relational structure $\mathcal{D} := \langle D, R_\sigma, R_\tau \rangle$, where $R_\sigma(d, e) \iff \sigma(d) = e$ and $R_\tau(d, e) \iff \tau(d) = e$.

The following lemma shows that the three representations are equivalent.

Lemma 3.4. *Let (G, π) and (G', π') be two embedded graphs, with associated relational structures $\mathcal{R}, \mathcal{D}$ and $\mathcal{R}', \mathcal{D}'$, respectively. Then $(G, \pi) \cong_R (G', \pi') \iff \mathcal{R} \cong \mathcal{R}' \iff \mathcal{D} \cong \mathcal{D}'$.*

Theorem 3.3 reduces a topological question to a combinatorial one. Lemma 3.4, in turn, allows us to represent an embedded graph (G, π) as a relational structure, its rotation structure $\mathcal{R}$ or, equivalently, its dart structure $\mathcal{D}$. We rely on these three equivalent representations throughout the rest of the paper.

[3]Note that our approaches cover planar embeddings, as they are equivalent to embeddings on the sphere.

4 THE CYCLIC WEISFEILER–LEMAN TEST

As discussed in the previous section, embedded graphs are fully described by the combinatorial information encoded by their rotation system. Color refinement, the standard heuristic for (abstract) graph isomorphism, can thus be adapted to the embedded setting. In this section, we introduce *Cyclic Weisfeiler–Leman* (CWL), a color refinement algorithm that aggregates the neighbors of each vertex in their cyclic order. We establish its basic properties and outline the proof of our main expressivity result: CWL identifies all embedded trees.

A classical result states that WL identifies all trees (Immerman & Lander, 1990; Arvind et al., 2017). We will show that proving an analogous result for embedded trees requires more effort. At the same time, the embedded setting is combinatorially richer than the abstract one: equipping a tree with cyclic orders at its vertices produces far more isomorphism classes[4]. Our result shows that incorporating these local cyclic orders into WL, by replacing the multiset of neighbor colors with their cyclically ordered tuple, yields an invariant that identifies every embedded tree.

Definition 4.1 (Cyclic Weisfeiler–Leman)**.** Let (G, π) be an embedded graph. Starting from a constant coloring $c_v^{(0)} = c^{(0)}$ on the vertices $v \in V(G)$, the *Cyclic Weisfeiler–Leman* (CWL) algorithm iteratively sets

$$c_v^{(t+1)} = \text{HASH}\left(c_v^{(t)}, (c_{v_1}^{(t)}, \ldots, c_{v_d}^{(t)})_\sim\right),$$

where $d = \deg(v)$, the degree of node v, $v_1, \ldots, v_d$ are the neighbors of v in the cyclic order π_v, and HASH is as above. $(\cdot)_\sim$ denotes equivalence under cyclic shifts, i.e. $(a_1, \ldots, a_d)_\sim = (a_{j+1}, \ldots, a_d, a_1, \ldots, a_j)_\sim$ for every $0 \le j < d$.

The fundamental difference from the usual WL algorithm is the aggregation: *the unordered multiset of neighbor colors is replaced by their tuple, cyclically ordered according to the embedding*. We state three basic properties of CWL. Proofs are deferred to Appendix B.

(i) At every iteration, CWL refines WL, assuming the same constant initial coloring.
(ii) CWL stabilizes. Writing c_v for the stable color, we define $\text{CWL}(G, \pi) := \{\{c_v : v \in V(G)\}\}$. We write $(G, \pi) \equiv_{\text{CWL}} (G', \pi')$ if $\text{CWL}(G, \pi) = \text{CWL}(G', \pi')$.
(iii) CWL is invariant under embedding equivalence: if $(G, \pi) \cong_R (G', \pi')$, then $\text{CWL}(G, \pi) = \text{CWL}(G', \pi')$.

Theorem 4.2 (CWL identifies all embedded trees)**.** *Let (T, π) and (G, π') be an embedded tree and an embedded graph, respectively, then*

$$(T, \pi) \equiv_{CWL} (G, \pi') \iff (T, \pi) \cong_R (G, \pi').$$

Although this result is a direct analog of the corresponding result for WL on trees, its proof is non-trivial and requires significant effort. For ordinary WL, the color of a vertex after t iterations determines its t-ball, which is defined as the subgraph induced by the vertices at distance at most t from v. This relies on the fact that a t-ball of a node is uniquely determined by the $(t-1)$-balls of its neighbors. In the presence of symmetries, the same argument may fail in the embedded setting, because different cyclic-shift alignments are possible between the smaller balls, producing non-equivalent embedded t-balls (see Figure 1). We resolve this ambiguity using *eccentricity* to identify the parent's position in the stable cyclic tuple.

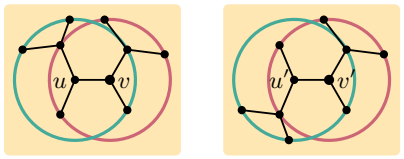

Figure 1: Example of non-equivalent embedded 3-balls, rooted at v, v' with $c_v^{(3)} = c_{v'}^{(3)}$. The radius-2 balls are equivalent, together with their intersections, but these local equivalences do not extend to the 3-balls.

The proof rests on two intermediate results. First, for trees, the stable WL color of a vertex determines its eccentricity. Second, eccentricity strictly increases along every path moving away from the central vertices. Together, these results imply a key structural fact: each non-central vertex can identify its parent, its unique neighbor on the path to the tree center, by its color. This fixes a starting point for

[4]For $n = 10, 20, 30$, the numbers of abstract trees are respectively in the order of 10^2, 10^5, and 10^{10}, whereas the corresponding numbers of embedded trees are 10^2, 10^7, and 10^{13} (Harary & Palmer, 1973).

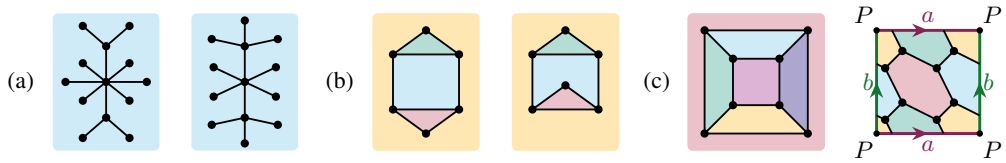

Figure 2: (a) Two non-equivalent embedded trees that 1-TWL$^{\mathcal{D}}$ fails to distinguish. (b) Two non-equivalent embeddings of the same planar graph that are distinguished by 1-TWL$^{\mathcal{D}}$ and not by CWL. (c) A planar and a toroidal embedding of the cube Q_3 that are not distinguished by 2-TWL$^{\mathcal{R}}$.

the cyclic order, allowing us to determine the cyclic-shift alignment and to construct the isomorphism that preserves the rotation system.

The central result of this section shows that CWL can identify all embedded trees. Given the parallelism to WL, a natural question arises: can CWL distinguish larger classes of embedded graphs? And if not, are higher-dimensional versions of our approach more powerful? Figure 2b shows a simple pair of planar graphs that CWL fails to distinguish. On the other hand, an extension to higher-dimensional variants of CWL is not easy to define, as there is no natural way to include the cyclic order for tuples of nodes. Therefore, as mentioned in Section 3, in the next section we switch from (G, π) to the relational structures $\mathcal{R}$ and $\mathcal{D}$.

5 The Topological Weisfeiler–Leman Tests

To overcome the shown limitations of CWL, we introduce our higher-dimensional *Topological WL* (TWL) algorithms based on the representation of embedded graphs as relational structures. We write k-TWL$^{\mathcal{R}}$ and k-TWL$^{\mathcal{D}}$ (k-TWL$^{\mathcal{R}}_{cf}$ and k-TWL$^{\mathcal{D}}_{cf}$) for k-WL (k-WL$_{cf}$) operating on the relational structures $\mathcal{R}$ and $\mathcal{D}$ defined in Section 3, respectively.

5.1 The WL on the Rotation Structure

We rely on the rotation structure $\mathcal{R}$ and write k-TWL$^{\mathcal{R}}$ (k-TWL$^{\mathcal{R}}_{cf}$) for k-WL (k-WL$_{cf}$) operating on $\mathcal{R}$. Grohe & Verbitsky (2006) studied this algorithm from a circuit-complexity perspective. Using the relationship between k-WL and pebble games from Theorem 3.2, they show that 4-TWL$^{\mathcal{R}}$ distinguishes all embedded graphs, i.e., Spoiler has a winning strategy in the 5-pebble game. We improve their result and prove that 3-TWL$^{\mathcal{R}}$ is equally expressive, by presenting a new winning strategy for Spoiler using 4 pebbles.

Theorem 5.1. *Let $\mathcal{R} = \langle V, E, T\rangle$ and $\mathcal{R}' = \langle V', E', T'\rangle$ be two non-isomorphic rotation structures and let $m = \min\{|E|, |E'|\}$. Then, Spoiler has a winning strategy in the 4-pebble game* $\mathrm{EF}^4(\mathcal{R}, \mathcal{R}')$ *using at most $4m + 2$ rounds.*

By Theorem 3.2, it then follows as a corollary that 3-TWL$^{\mathcal{R}}$ is complete for embedded graphs.

Corollary 5.2. *For any two rotation structures $\mathcal{R}$ and $\mathcal{R}'$,*

$$\mathcal{R} \equiv_{3\text{-TWL}^{\mathcal{R}}} \mathcal{R}' \iff \mathcal{R} \equiv_{3\text{-TWL}^{\mathcal{R}}_{\mathrm{cf}}} \mathcal{R}' \iff \mathcal{R} \cong \mathcal{R}'.$$

We note that the result by Grohe & Verbitsky (2006) using 5 pebbles achieves a Spoiler win in a logarithmic number of rounds, which is essential for their circuit complexity setting, while Theorem 5.1 requires a linear number of rounds. However, the number of rounds played in the game affects neither the expressivity analysis nor the running time of k-TWL$^{\mathcal{R}}$, when using the optimized implementation by Immerman & Sengupta (2019), which computes the stable coloring in time $O(k^2 n^{k+1} \log n)$. We complement Corollary 5.2 by showing that 2-TWL$^{\mathcal{R}}$ does not distinguish all embedded graphs.

Proposition 5.3. *2-TWL$^{\mathcal{R}}$ cannot distinguish the planar and the toroidal embeddings of the cube in Figure 2c.*

Together, our results thus show that the WL dimension of embedded graphs is exactly 3. In the rest of this section, we give an overview of the proof of Theorem 5.1. The full proof, and a generalization of the theorem where $\mathcal{R}'$ is not required to be a rotation structure, can be found in Appendix C.2.

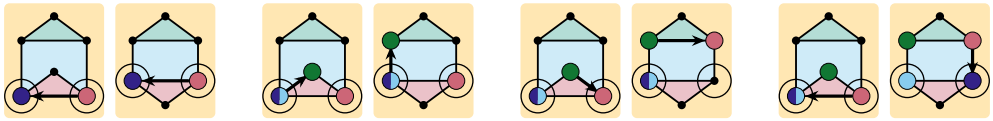

Figure 3: Spoiler's strategy in the proof of Theorem 5.1. He chooses the left side and traces $(\tau\sigma)^3$.

Let $\mathcal{W}$ be the set of finite words over $\{\sigma, \sigma^{-1}, \tau\}$. For $w \in \mathcal{W}$ and a dart $d \in D$, we write d^w for the dart obtained by applying the permutations corresponding to the symbols of w successively from left to right. A word w is a *loop* at dart d if $d^w = d$. Spoiler's winning strategy will be to pick a suitable word w and *trace* it from a starting dart d by iteratively putting pebbles on the endpoints of the dart obtained by applying the next letter from w. We will show that Spoiler tracing a word forces all moves by Duplicator. The following lemma shows Spoiler which word to trace.

Lemma 5.4. *Let $\mathcal{D} = \langle D, R_\sigma, R_\tau \rangle$ and $\mathcal{D}' = \langle D', R'_\sigma, R'_\tau \rangle$ be two non-isomorphic dart structures, let $m = \min\{|D|, |D'|\}/2$, and let $d_0 \in D$ and $d'_0 \in D'$. Then, there exists a word $w \in \mathcal{W}$ of length at most $4m - 1$ such that w is a loop at exactly one of d_0 and d'_0.*

Spoiler can win the game by first placing pebbles on an arbitrary dart d and then tracing the word w obtained from Lemma 5.4 using 3 pebbles, keeping the fourth pebble on an endpoint of d. After tracing w, exactly one of d and d' will have an endpoint with two pebbles placed on it, as w is a loop on exactly one of them. Thus, the pebbles will not form a partial isomorphism, and Spoiler wins. See Appendix C.2 for the proof and Figure 3 for a visualization of the game.

5.2 The WL on the Dart Structure

While 3-TWL$^{\mathcal{R}}$ is complete on embedded graphs, it operates on vertex triples and can be time-expensive in practice. In this section, we consider k-WL on the dart structure, denoted k-TWL$^{\mathcal{D}}$, and prove its completeness at dimension 2. For sparse graphs of $O(n)$ edges, e.g., graphs embedded on surfaces of bounded genus, this improves the runtime complexity by a factor of n. Furthermore, the representation of embedded graphs by the permutations σ, τ on darts is closely related to the Doubly Connected Edge List (DCEL) data structure used to represent graph embeddings in the graph drawing literature (Muller & Preparata, 1978). The following theorem is our main result of this section.

Theorem 5.5. *Let $\mathcal{D} = \langle D, R_\sigma, R_\tau \rangle$ and $\mathcal{D}' = \langle D', R'_\sigma, R'_\tau \rangle$ be two non-isomorphic dart structures and let $m = \min\{|D|, |D'|\}/2$. Then Spoiler has a winning strategy in the 3-pebble game $\mathrm{EF}^3(\mathcal{D}, \mathcal{D}')$ using at most $4m + 1$ rounds.*

The proof of Theorem 5.5 is similar to that of Theorem 5.1 and can be found in Appendix C.4. By Theorem 3.2, 2-TWL$^{\mathcal{D}}$ on the dart structure is complete.

Corollary 5.6. *For any two dart structures $\mathcal{D}$ and $\mathcal{D}'$,*

$$\mathcal{D} \equiv_{\text{2-TWL}^{\mathcal{D}}} \mathcal{D}' \iff \mathcal{D} \equiv_{\text{2-TWL}^{\mathcal{D}}_{\text{cf}}} \mathcal{D}' \iff \mathcal{D} \cong \mathcal{D}'.$$

Analogous to Proposition 5.3, we show that Corollary 5.6 is tight in the sense that 1-TWL$^{\mathcal{D}}$ is not complete. In particular, we show that it is not even complete on trees.

Proposition 5.7. *1-TWL$^{\mathcal{D}}$ cannot distinguish the two embedded trees in Figure 2a.*

5.3 Summary of our Theoretical Contribution

In the previous sections, we presented the theoretical results concerning our three methods. In particular, the completeness of 3-TWL$^{\mathcal{R}}$ and 2-TWL$^{\mathcal{D}}$, together with the examples on which their lower-dimensional variants fail, tightly characterize their expressivity with respect to their dimension. It is now natural to ask: how do the different methods compare in terms of expressive power? The failure of 1-TWL$^{\mathcal{D}}$, CWL and 2-TWL$^{\mathcal{R}}$ on the pairs in Figures 2a, 2b and 2c, respectively, already provides a direct comparison with the complete methods. Furthermore, a direct computation shows that 1-TWL$^{\mathcal{D}}$ distinguishes the pair in Figure 2b, which CWL cannot. These results, together with Theorem 4.2, show that CWL and 1-TWL$^{\mathcal{D}}$ are incomparable. In Appendix D we complete our expressivity comparison, by showing that 2-TWL$^{\mathcal{R}}$ is strictly more expressive than the 1-dimensional methods. We summarize our full description in Table 1 and in the following theorem:

Theorem 5.8. $\{1\text{-}TWL^{\mathcal{D}} \not\simeq CWL\} \prec 2\text{-}TWL^{\mathcal{R}} \prec \{3\text{-}TWL^{\mathcal{R}} \simeq 2\text{-}TWL^{\mathcal{D}}\}$.

One may ask why we do not consider 1-TWL$^{\mathcal{R}}$. The answer lies in the fact that to fully account for a $(k+1)$-ary relation, we need at least k-WL, which incorporates the atomic types of $(k+1)$-tuples during the aggregation (see Remark 3.1). Thus, our ternary relation T requires at least dimension 2.

6 Experiments

In the previous sections, we established the expressivity of our methods. However, whether these theoretical results translate into practical benefits remains an open question. We address this question through an empirical evaluation of novel GNNs built upon our WL algorithms. Specifically, we ask:

(Q1) Does the expressivity of our GNNs align in practice with that of the respective WL methods?
(Q2) Are our GNNs able to learn topological properties?
(Q3) Are our GNNs competitive in tasks involving 3D geometry?

To address (Q1), we consider a synthetic dataset of embedded graphs, and to address (Q2), we use both the same dataset and MANTRA (Ballester et al., 2025), a benchmark dataset for topological property predictions. To address (Q3), we consider ChIRo (Adams et al., 2022), a standard benchmark dataset for chirality-sensitive tasks. Our models are highlighted in purple.

Models.[5] We introduce two GNN architectures: *Cyclic Graph Network* (CGN) and *Topological Graph Network* (TGN). CGN augments GIN (Xu et al., 2019) with the cyclic-shift invariant aggregation of CWL, implemented through windows of length k in the style of ChiENN (Gainski et al., 2023). TGN augments the 2-WL-based architecture PPGN (Maron et al., 2019) with the ternary relation T of 2-TWL$^{\mathcal{R}}$. See Appendix E for more details on the architectures.

Expressivity (Q1). We generate 21k cellular embeddings of 230 graphs, on surfaces of genus between 0 and 5 (details in Appendix F), and collect pairs of non-equivalent embeddings of the same graph with matching genus. We group these pairs into sets $\mathcal{G}_{\text{CWL}}$, $\mathcal{G}_{2\text{-TWL}^{\mathcal{R}}}$, $\mathcal{G}_{3\text{-TWL}^{\mathcal{R}}}$, by the weakest test that separates them: CWL, 2-TWL$^{\mathcal{R}}$, or only the complete refinements. The task is to identify the equivalence class of an instance with randomly relabeled nodes. Table 2 shows that the learned models closely reproduce the expressivity of their WL counterparts. On $\mathcal{G}_{2\text{-TWL}^{\mathcal{R}}}$, only TGN succeeds (97.7%), reflecting the gap between CWL and 2-TWL$^{\mathcal{R}}$. No model exceeds 50% accuracy on $\mathcal{G}_{3\text{-TWL}^{\mathcal{R}}}$, as our theory predicts.

Table 2: Synthetic benchmarks: test accuracy (%). Task 1: identify the equivalence class of a randomly relabeled embedded graph, grouped by the weakest distinguishing test. Task 2: predict the genus on unseen graphs.

Model	$\mathcal{G}_{\text{CWL}}$	$\mathcal{G}_{2\text{-TWL}^{\mathcal{R}}}$	$\mathcal{G}_{3\text{-TWL}^{\mathcal{R}}}$	Genus
GIN	27.6	11.4	50.0	41.6
PPGN	26.3	19.3	50.0	42.1
CGN	100.0	11.1	50.0	52.0
TGN	100.0	97.7	50.0	92.8

Topological property prediction (Q2). MANTRA (Ballester et al., 2025) contains 43k triangulations of closed surfaces on at most 10 vertices, each given as a list of triangles over indexed vertices, and poses three embedding-dependent prediction tasks: orientability, Betti numbers, and homeomorphism type of the surface. We extract the rotation system from the given triangulation in the following way: we consider the vertex link of each node, i.e., the cycle graph of its neighbors, which is encoded by the triangulation, and we give it a direction that depends on the vertex indices. We train for either 6 or 100 epochs: TGN reaches perfect accuracy on every task. In Appendix G.5, we provide further experiments supporting the strong performance of our models. Table 3a presents the results for the prediction of the orientability of the triangulation, reported to be the most difficult task in the original benchmark. We report the best-performing existing method, CellMP[6] (Bodnar et al., 2021a), which achieves 65% balanced accuracy at 50× the runtime of our model. As a second task, we predict the genus of all the synthetic embeddings used for **(Q1)**, split by graph so that no test graph occurs in training (see Appendix F for more details). Here, the inputs are rotation systems, so the genus is determined by the embedding, whereas the same graph can be embedded on surfaces of different genera. While graph-only models remain at 42%, CGN and TGN read the rotation system and reach 52.0% and 92.8% (see Table 2). In particular, TGN generalizes to embeddings of unseen graphs.

[5] Our anonymized implementation is available at anonymous.4open.science/r/Topological-WL-A603.

[6] We reran CellMP with the released code but could not reproduce the published score. Both our and the published results are in Table 3a.

Table 3: **(a)** We report balanced accuracy[7], AUROC and accuracy, all in %, and runtime. **(b)** We report accuracy on the two tasks R/S and Ranking, and training time for Ranking. In both tables, results are taken from the literature unless marked with †. Runtimes are measured from our reruns.

(a) MANTRA orientability prediction

Model	Bal. acc. ↑	AUROC ↑	Acc. ↑	s/ep
CellMP	65±7	–	64±26	–
CellMP†	52±2	84±8	92±0	140
GIN	53±1	89±0	93±0	1.5
PPGN	58±4	90±0	92±0	2.5
CGN	50±0	89±0	92±0	1.4
TGN	100±0	100±0	100±0	2.8

(b) ChIRo molecular chirality

Model	R/S (%) ↑	Ranking (%) ↑	s/ep
ChIRo	98.5±0.2	72.0±0.5	25.8
ChiDeK	99.8±0.1	72.8±0.2	223
ChiENN†	98.6±0.2	73.7±0.6	38.4
ChiENN	98.9±0.0	76.0±0.2	268
CGN	98.9±0.2	75.4±0.6	6.1
TGN	98.6±0.2	73.1±0.3	45.2

3D Geometry: chirality (Q3). ChIRo (Adams et al., 2022) is a standard benchmark dataset for chirality-aware models. It consists of pairs of enantiomers, molecules that are non-superimposable mirror images of each other, each with one labeled stereocenter and several conformers. ChIRo poses two tasks: R/S classification (on 39k pairs) and Ranking (on 35k pairs). The first is to classify the stereocenter as R or S, providing a basic test of chirality sensitivity. The second is to rank the two enantiomers of a pair by their docking scores against a target protein. We construct the rotation system by cyclically ordering the bonds at each atom using their directions and RDKit's canonical atom ranking (Schneider et al., 2015) (Appendix H). We compare against three state-of-the-art chirality-aware models, ChIRo (Adams et al., 2022), ChiENN (Gainski et al., 2023) and ChiDeK (Shi et al., 2026). All of them solve R/S classification (Table 3b). On ranking, CGN reaches 75.4%, above ChiDeK (72.8%) and ChIRo (72.0%). ChiENN reports an accuracy of 76.0% with regenerated conformers and random-walk positional encodings. For a further comparison, we also re-run its released model on the dataset conformers and without the encoding (ChiENN†, in the table), where it achieves 73.7%. Note that we compare our models to highly specialized chirality-aware methods of varying complexity. Our models are not only simpler, which is reflected in the runtime (6.1 seconds per epoch against 25.8 for ChIRo, 38.4 for ChiENN†, 223 for ChiDeK, and 268 for ChiENN), but also more general as they can be applied to other topology-aware tasks.

Discussion. In the expressivity experiments, our models closely follow the theory: TGN succeeds up to $\mathcal{G}_{2\text{-TWL}^{\mathcal{R}}}$ and CGN up to $\mathcal{G}_{\text{CWL}}$. The same gap persists in the topological predictions, where only TGN solves orientability and generalizes to genus. On chirality, CGN outperforms TGN, although 2-TWL$^{\mathcal{R}}$ is more expressive than CWL. A possible reading is that the dense global aggregation of TGN suits topological invariants, which are global, while the sparse local aggregation of CGN suits chirality, a local property of the cyclic order at the stereocenter.

7 Conclusions and Future Work

Our work connects topological graph theory with graph machine learning and is, to our knowledge, the first to study message passing on embedded graphs from this point of view; many questions remain open. The expressive power of the non-complete methods CWL, 1-TWL$^{\mathcal{D}}$ and 2-TWL$^{\mathcal{R}}$ is not yet characterized, and a description of our methods in terms of homomorphism counts, as known for abstract graphs, is missing. Finally, our methods capture embeddings up to orientation-preserving homeomorphism; addressing the stronger notion of ambient isotopy is a challenge we leave for future work.

Reproducibility Statement

Our implementation is available at anonymous.4open.science/r/Topological-WL-A603. Appendices E, F, G and H give details about the datasets, our models and the experimental setup.

[7]While MANTRA claims to report AUROC, their public implementation computes balanced accuracy, i.e., AUROC after rounding the predictions to a binary classification.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic datasets, providing critical ingredients for proving mathematical claims, assisting in proof writing, implementing our methods, and designing experiments. We have not used generative AI tools for developing theoretical models or frameworks, for formulating mathematical claims, for proposing or refining hypotheses, or for interpreting results. Translation, cleaning, and reformatting datasets and qualitative data analysis are not applicable to this work. Additionally, we used generative AI tools for creating software code, identifying relevant literature, and drafting parts of the appendix. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Keir Adams, Lagnajit Pattanaik, and Connor W. Coley. Learning 3d representations of molecular chirality with invariance to bond rotations. In *ICLR*, 2022.

Amirreza Akbari, Amauri H. Souza, and Vikas K. Garg. The logical expressiveness of topological neural networks. In *ICLR*, 2026.

Vikraman Arvind, Johannes Köbler, Gaurav Rattan, and Oleg Verbitsky. Graph isomorphism, color refinement, and compactness. *Comput. Complex.*, 26(3):627–685, 2017.

Rubén Ballester and Bastian Rieck. On the expressivity of persistent homology in graph learning. In *LoG*, volume 269 of *Proceedings of Machine Learning Research*, pp. 42. PMLR, 2024.

Rubén Ballester, Ernst Röell, Daniel Bin Schmid, Mathieu Alain, Sergio Escalera, Carles Casacuberta, and Bastian Rieck. MANTRA: the manifold triangulations assemblage. In *ICLR*, 2025.

Claudio Battiloro, Ege Karaismailoglu, Mauricio Tec, George Dasoulas, Michelle Audirac, and Francesca Dominici. E(n) equivariant topological neural networks. In *ICLR*, 2025.

Franka Bause, Fabian Jogl, Patrick Indri, Tamara Drucks, David Penz, Nils Morten Kriege, Thomas Gärtner, Pascal Welke, and Maximilian Thiessen. Maximally expressive gnns for outerplanar graphs. *Trans. Mach. Learn. Res.*, 2025, 2025.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Liò, Guido F. Montúfar, and Michael M. Bronstein. Weisfeiler and lehman go cellular: CW networks. In *NeurIPS*, pp. 2625–2640, 2021a.

Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F. Montúfar, Pietro Lió, and Michael M. Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1026–1037. PMLR, 2021b.

Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Stefanos Zafeiriou, and Michael M. Bronstein. Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In *ICCV*, pp. 7212–7221. IEEE, 2019.

Gunnar Brinkmann and Brendan D. McKay. Fast generation of planar graphs. *MATCH Communications in Mathematical and in Computer Chemistry*, 58(2):323–357, 2007.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Velickovic. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *CoRR*, abs/2104.13478, 2021.

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

Radoslav Dimitrov, Zeyang Zhao, Ralph Abboud, and .Ismail .Ilkan Ceylan. Plane: Representation learning over planar graphs. In *NeurIPS*, 2023.

Stefania Ebli, Michaël Defferrard, and Gard Spreemann. Simplicial neural networks. *NeurIPS Workshop on Topological Data Analysis and Beyond*, 2020.

John Robert Edmonds Jr. *A combinatorial representation for oriented polyhedral surfaces*. PhD thesis, University of Maryland, 1960.

Floor Eijkelboom, Rob Hesselink, and Erik J. Bekkers. E(n) equivariant message passing simplicial networks. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pp. 9071–9081. PMLR, 2023.

Yam Eitan, Yoav Gelberg, Guy Bar-Shalom, Fabrizio Frasca, Michael M. Bronstein, and Haggai Maron. Topological blindspots: Understanding and extending topological deep learning through the lens of expressivity. In *ICLR*. OpenReview.net, 2025.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *CoRR*, abs/1903.02428, 2019.

Piotr Gainski, Michal Koziarski, Jacek Tabor, and Marek Smieja. Chienn: Embracing molecular chirality with graph neural networks. In *ECML/PKDD (3)*, volume 14171 of *Lecture Notes in Computer Science*, pp. 36–52. Springer, 2023.

Vikas K. Garg, Stefanie Jegelka, and Tommi S. Jaakkola. Generalization and representational limits of graph neural networks. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3419–3430. PMLR, 2020.

Martin Grohe. Isomorphism testing for embeddable graphs through definability. In *STOC*, pp. 63–72. ACM, 2000.

Martin Grohe and Sandra Kiefer. A linear upper bound on the weisfeiler-leman dimension of graphs of bounded genus. In *ICALP*, volume 132 of *LIPIcs*, pp. 117:1–117:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

Martin Grohe and Oleg Verbitsky. Testing graph isomorphism in parallel by playing a game. In *ICALP (1)*, volume 4051 of *Lecture Notes in Computer Science*, pp. 3–14. Springer, 2006.

Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, Nina Miolane, Aldo Guzmán-Sáenz, Karthikeyan Natesan Ramamurthy, Tolga Birdal, Tamal K. Dey, Soham Mukherjee, Shreyas N. Samaga, Neal Livesay, Robin Walters, Paul Rosen, and Michael T. Schaub. Topological deep learning: Going beyond graph data. *arXiv preprint arXiv:2206.00606*, 2022.

Frank Harary and Edgar M. Palmer. *Graphical enumeration*. Addison-Wesley, 1973.

Christoph D. Hofer, Florian Graf, Bastian Rieck, Marc Niethammer, and Roland Kwitt. Graph filtration learning. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4314–4323. PMLR, 2020.

Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten M. Borgwardt. Topological graph neural networks. In *ICLR*. OpenReview.net, 2022.

Neil Immerman. *Descriptive complexity*. Graduate texts in computer science. Springer, 1999.

Neil Immerman and Eric Lander. Describing graphs: A first-order approach to graph canonization. In *Complexity Theory Retrospective: In Honor of Juris Hartmanis on the Occasion of His Sixtieth Birthday, July 5, 1988*, pp. 59–81. Springer, 1990.

Neil Immerman and Rik Sengupta. The k-dimensional weisfeiler-leman algorithm. *CoRR*, abs/1907.09582, 2019.

Chaitanya K. Joshi, Cristian Bodnar, Simon V. Mathis, Taco Cohen, and Pietro Lio. On the expressive power of geometric graph neural networks. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pp. 15330–15355. PMLR, 2023.

Sandra Kiefer, Ilia Ponomarenko, and Pascal Schweitzer. The weisfeiler-leman dimension of planar graphs is at most 3. *J. ACM*, 66(6):44:1–44:31, 2019.

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. In *NeurIPS*, pp. 2153–2164, 2019.

David E. Muller and Franco P. Preparata. Finding the intersection of two convex polyhedra. *Theor. Comput. Sci.*, 7:217–236, 1978.

Theodore Papamarkou, Tolga Birdal, Michael M. Bronstein, Gunnar E. Carlsson, Justin Curry, Yue Gao, Mustafa Hajij, Roland Kwitt, Pietro Lio, Paolo Di Lorenzo, Vasileios Maroulas, Nina Miolane, Farzana Nasrin, Karthikeyan Natesan Ramamurthy, Bastian Rieck, Simone Scardapane, Michael T. Schaub, Petar Velickovic, Bei Wang, Yusu Wang, Guo-Wei Wei, and Ghada Zamzmi. Position: Topological deep learning is the new frontier for relational learning. In *ICML*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39529–39555. PMLR / OpenReview.net, 2024.

Mathilde Papillon, Sophia Sanborn, Mustafa Hajij, and Nina Miolane. Architectures of topological deep learning: A survey on topological neural networks. *CoRR*, abs/2304.10031, 2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pp. 8024–8035, 2019.

T. Mitchell Roddenberry, Nicholas Glaze, and Santiago Segarra. Principled simplicial neural networks for trajectory prediction. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9020–9029. PMLR, 2021.

Ryoma Sato, Makoto Yamada, and Hisashi Kashima. Approximation ratios of graph neural networks for combinatorial problems. In *NeurIPS*, pp. 4083–4092, 2019.

Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9323–9332. PMLR, 2021.

Nadine Schneider, Roger A. Sayle, and Gregory A. Landrum. Get your atoms in order - an open-source implementation of a novel and robust molecular canonicalization algorithm. *J. Chem. Inf. Model.*, 55(10):2111–2120, 2015.

Runhan Shi, Zhicheng Zhang, Letian Chen, Gufeng Yu, and Yang Yang. Learning molecular chirality via chiral determinant kernels. In *ICLR*, 2026.

Thom Sulanke. Generating irreducible triangulations of surfaces. *arXiv preprint math/0606687*, 2006.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

Qi Zhao, Ze Ye, Chao Chen, and Yusu Wang. Persistence enhanced graph neural network. In *AISTATS*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2896–2906. PMLR, 2020.

A THE WL ALGORITHM

We say that B is at least as expressive as A, written $A \preceq B$, if $B(x) = B(y)$ implies $A(x) = A(y)$ for all $x, y \in X$; the two are equivalent, $A \simeq B$, if each is at least as expressive as the other, B is strictly more expressive, $A \prec B$, if $A \preceq B$ but not $B \preceq A$, and they are incomparable, $A \not\simeq B$, if neither is at least as expressive as the other.

A *coloring* of a finite set X is a map c from X to a set of colors and induces a partition of X into color classes. A coloring c' *refines* c if $c'(x) = c'(y) \implies c(x) = c(y)$ for all $x, y \in X$. For an iterative color refinement algorithm, we denote its coloring at iteration t by $c^{(t)}$. The algorithm *stabilizes* at the first iteration T for which $c^{(T)}$ and $c^{(T+1)}$ induce the same partition. We call $c^{(T)}$ its *stable coloring*.

Let $G = (V, E)$ be a graph with vertex set V and edge set E. Starting from a constant coloring $c_v^{(0)} = c^{(0)}$ on the vertices $v \in V$, the Weisfeiler–Leman algorithm (WL) iteratively sets:

$$c_v^{(t+1)} = \textsc{hash}\left(c_v^{(t)}, \{\{c_u^{(t)} : u \in N(v)\}\}\right),$$

where $\{\{\cdot\}\}$ denotes a multiset, HASH is an injective map to a color, and $N(v)$ denotes the set of neighbors of node v. The same HASH function is used within and across graphs, and two graphs are always compared by running the refinement on their disjoint union. All our tests follow this convention. Writing c_v for the stable color, we define $\mathrm{WL}(G) \coloneqq \{\{c_v : v \in V\}\}$ and write $G \equiv_{\mathrm{WL}} G'$ if $\mathrm{WL}(G) = \mathrm{WL}(G')$. We say that WL distinguishes G and G' if $G \not\equiv_{\mathrm{WL}} G'$.

B RESULTS ON CWL

In this section, we prove the three basic properties (i)–(iii) of CWL stated after Definition 4.1, and Theorem 4.2.

B.1 BASIC PROPERTIES

Lemma B.1 (CWL refines WL)**.** *Let (G, π) be an embedded graph. For every $t \geq 0$ and every pair $u, v \in V(G)$,*

$$c_u^{CWL,(t)} = c_v^{CWL,(t)} \implies c_u^{WL,(t)} = c_v^{WL,(t)}.$$

Proof. We prove by induction on t the stronger claim that the CWL color determines the WL color: there exists $\Phi_t \colon \mathcal{C} \to \mathcal{C}$ with $c_v^{\mathrm{WL},(t)} = \Phi_t(c_v^{\mathrm{CWL},(t)})$ for every $v \in V(G)$.

Base case ($t = 0$). Both algorithms assign the same constant initial color, so $\Phi_0 = \mathrm{id}$.

Inductive step. Assume Φ_{t-1} exists. By injectivity of the CWL hash, $c_v^{\mathrm{CWL},(t)}$ determines

$$\big(c_v^{\mathrm{CWL},(t-1)},\ (c_{v_1}^{\mathrm{CWL},(t-1)}, \ldots, c_{v_k}^{\mathrm{CWL},(t-1)})_{\sim}\big).$$

A cyclic tuple determines its underlying multiset, and applying Φ_{t-1} pointwise converts these data into

$$\big(c_v^{\mathrm{WL},(t-1)},\ \{\{c_{v_1}^{\mathrm{WL},(t-1)}, \ldots, c_{v_k}^{\mathrm{WL},(t-1)}\}\}\big),$$

which is the input to the WL hash producing $c_v^{\mathrm{WL},(t)}$. The composition of the described maps yields Φ_t. □

From now on, $c_v^{(t)}$ denotes the color of vertex v at iteration t computed by CWL. When we need to distinguish between the two coloring schemes, we write $c_v^{\mathrm{CWL},(t)}$ and $c_v^{\mathrm{WL},(t)}$.

Lemma B.2 (CWL stabilizes)**.** *Let (G, π) be an embedded graph with $n = |V(G)|$. The sequence of colorings $(c^{(t)})_{t \geq 0}$ produced by CWL stabilizes, i.e., there exists $t_0 \leq n - 1$ such that, for all $t \geq t_0$ and all $u, v \in V(G)$,*

$$c_u^{(t)} = c_v^{(t)} \iff c_u^{(t_0)} = c_v^{(t_0)}.$$

Proof. For each $t \geq 0$, the coloring $c^{(t)}$ induces a partition P_t of $V(G)$, where u and v belong to the same block if and only if $c_u^{(t)} = c_v^{(t)}$. Since HASH is injective and takes $c_v^{(t-1)}$ as its first argument, equal colors at iteration t imply equal colors at iteration $t-1$, so P_{t+1} refines P_t for all $t \geq 0$. As $V(G)$ is finite, at most $n-1$ strict refinements can occur, so there exists $t_0 \leq n-1$ with $P_{t_0} = P_{t_0+1}$.

It remains to show that $P_{t_0} = P_{t_0+1}$ implies $P_{t_0+1} = P_{t_0+2}$. Since $P_{t_0} = P_{t_0+1}$, we have $c_u^{(t_0)} = c_v^{(t_0)} \iff c_u^{(t_0+1)} = c_v^{(t_0+1)}$ for all u, v. This makes the map β sending $c_w^{(t_0)} \mapsto c_w^{(t_0+1)}$ a well-defined bijection between the sets of colors occurring at iterations t_0 and t_0+1. Since $c_w^{(t_0+1)} = \beta(c_w^{(t_0)})$ for all w, we can rewrite the iteration t_0+2 as

$$c_v^{(t_0+2)} = \text{HASH}\big(\beta(c_v^{(t_0)}),\, (\beta(c_{v_1}^{(t_0)}), \ldots, \beta(c_{v_k}^{(t_0)}))_\sim\big).$$

Then the following chain of equivalences holds:

$$\begin{aligned}
c_u^{(t_0+2)} = c_v^{(t_0+2)} &\iff \begin{cases} \beta(c_u^{(t_0)}) = \beta(c_v^{(t_0)}) \\ (\beta(c_{u_1}^{(t_0)}), \ldots)_\sim = (\beta(c_{v_1}^{(t_0)}), \ldots)_\sim \end{cases} && \text{(injectivity of HASH)} \\
&\iff \begin{cases} c_u^{(t_0)} = c_v^{(t_0)} \\ (c_{u_1}^{(t_0)}, \ldots)_\sim = (c_{v_1}^{(t_0)}, \ldots)_\sim \end{cases} && (\beta \text{ bijective}) \\
&\iff c_u^{(t_0+1)} = c_v^{(t_0+1)} && \text{(injectivity of HASH)}
\end{aligned}$$

Therefore $P_{t_0+1} = P_{t_0+2}$. □

Consequently, stable CWL colors refine stable WL colors: apply Lemma B.1 at any iteration after which both algorithms have stabilized.

Lemma B.3 (CWL is invariant under equivalence of embeddings)**.** *Let (G, π) and (G', π') be two embedded graphs. If $(G, \pi) \cong_R (G', \pi')$, then CWL$(G, \pi)$ = CWL(G', π').*

Proof. By definition of $\cong_R$, there exists a graph isomorphism $\varphi\colon G \to G'$ that commutes with π and π'.

We want to show that, for every $t \geq 0$ and every $v \in V(G)$,

$$c_v^{(t)} = c_{\varphi(v)}^{(t)},$$

where the left-hand side is computed on (G, π) and the right-hand side on (G', π'), using the same HASH function.

Base case ($t = 0$). Both vertices receive the constant initial color $c^{(0)}$.

Inductive step. Assume $c_w^{(t-1)} = c_{\varphi(w)}^{(t-1)}$ for every $w \in V(G)$. Let $v \in V(G)$ have neighbors $(v_1, \ldots, v_k)$ in cyclic order under π. Since φ preserves the rotation system, the neighbors of $\varphi(v)$ in cyclic order under π' are $(\varphi(v_1), \ldots, \varphi(v_k))$. By the inductive hypothesis applied to v and to each v_i,

$$c_v^{(t-1)} = c_{\varphi(v)}^{(t-1)} \quad \text{and} \quad c_{v_i}^{(t-1)} = c_{\varphi(v_i)}^{(t-1)} \text{ for } i = 1, \ldots, k.$$

Hence, the cyclic tuples agree componentwise, and in particular as cyclic shift equivalence classes:

$$(c_{v_1}^{(t-1)}, \ldots, c_{v_k}^{(t-1)})_\sim = (c_{\varphi(v_1)}^{(t-1)}, \ldots, c_{\varphi(v_k)}^{(t-1)})_\sim.$$

Applying HASH to identical inputs yields $c_v^{(t)} = c_{\varphi(v)}^{(t)}$, completing the induction.

Choose t_0 large enough that both colorings have stabilized (Lemma B.2). Since φ is a bijection, $|V(G)| = |V(G')| = n$, so $t_0 \leq n-1$ suffices for both. Then $c_v = c_{\varphi(v)}$ for every $v \in V(G)$, and since $\varphi\colon V(G) \to V(G')$ is a bijection,

$$\text{CWL}(G, \pi) = \{\{c_v : v \in V(G)\}\} = \{\{c_{\varphi(v)} : v \in V(G)\}\} = \{\{c_{v'} : v' \in V(G')\}\} = \text{CWL}(G', \pi').$$

□

B.2 STANDARD ECCENTRICITY RESULTS

We collect the standard facts about eccentricity in trees that the proof of Theorem 4.2 uses.

Definition B.4 (Eccentricity, radius, center)**.** Let G be a finite connected graph and d its graph distance. The *eccentricity* of a vertex $v \in V(G)$ is

$$\mathrm{ecc}(v) = \max_{u \in V(G)} d(v, u).$$

The *radius* of G is $\mathrm{rad}(G) = \min_{v \in V(G)} \mathrm{ecc}(v)$, and the *center* of G is the set $C(G) = \{v \in V(G) : \mathrm{ecc}(v) = \mathrm{rad}(G)\}$.

Remark B.5. For any tree T with $|V(T)| \geq 3$, let $\hat{T} = T - L(T)$, where $L(T)$ is the set of leaves. $\hat{T}$ is a nonempty tree, and every leaf of T is adjacent to $\hat{T}$. Every eccentricity in T is realized by a leaf, so $\mathrm{ecc}_T(v) \leq \mathrm{ecc}_{\hat{T}}(v) + 1$ for all $v \in V(\hat{T})$. Conversely, a longest path from v in $\hat{T}$ ends at a vertex of degree at most 1 in $\hat{T}$ but at least 2 in T (or at v itself, if $\hat{T}$ is a singleton), hence it extends by a leaf of T. Thus $\mathrm{ecc}_T(v) = \mathrm{ecc}_{\hat{T}}(v) + 1$. For a leaf ℓ with neighbor p, every path from ℓ passes through p, and p has a neighbor other than ℓ, so $\mathrm{ecc}_T(\ell) = \mathrm{ecc}_T(p) + 1$. In particular, no leaf is central. Consequently $\mathrm{rad}(T) = \mathrm{rad}(\hat{T}) + 1$ and $C(T) = C(\hat{T})$.

Definition B.6 (Rooted tree)**.** Let T be a finite tree and $U \subseteq V(T)$ a non-empty set of vertices such that the subgraph induced by U is connected. We say T is *rooted at* U when we equip it with the *parent function* $\mathrm{par}\colon V(T) \setminus U \to V(T)$ defined as follows: for each $v \notin U$, $\mathrm{par}(v)$ is the unique neighbor of v lying on the path from v to U. If w is a neighbor of v with $\mathrm{par}(w) = v$, we call w a *child* of v.

The case $U = \{r\}$ recovers the usual notion of a tree rooted at a single vertex r, while the case $U = C$ (by Jordan's theorem, the center C is either a single vertex or a pair of adjacent vertices) yields the tree *rooted at its center*.

Lemma B.7. *Let T be a finite tree with center $C \subseteq V(T)$. For every $v \in V(T)$,*

$$\mathrm{ecc}(v) = \mathrm{rad}(T) + d(v, C),$$

where $d(v, C) = \min_{x \in C} d(v, x)$.

Proof. By induction on $|V(T)|$. The cases $|V(T)| = 1, 2$ are immediate.

For $|V(T)| \geq 3$, let $L \subseteq V(T)$ be the set of leaves and set $\hat{T} = T - L$. Since $|V(T)| \geq 3$, $\hat{T}$ is a non-empty tree, and by Remark B.5 $C(\hat{T}) = C(T) =: C$ and $\mathrm{rad}(\hat{T}) = \mathrm{rad}(T) - 1$.

For every non-leaf vertex $v \in V(\hat{T})$, the eccentricity of v in T is realized by some leaf of T (any longest path from v ends at a leaf), so $\mathrm{ecc}_T(v) = \mathrm{ecc}_{\hat{T}}(v) + 1$. Distances within $V(\hat{T})$ coincide in T and $\hat{T}$, so $d_T(v, C) = d_{\hat{T}}(v, C)$. The inductive hypothesis applied to $\hat{T}$ gives

$$\mathrm{ecc}_T(v) = \mathrm{ecc}_{\hat{T}}(v) + 1 = \mathrm{rad}(\hat{T}) + d_{\hat{T}}(v, C) + 1 = \mathrm{rad}(T) + d_T(v, C).$$

For a leaf $\ell \in L$, let p be its unique neighbor. Then $p \in V(\hat{T})$ (since $|V(T)| \geq 3$ forces p to be non-leaf), the path from ℓ to C passes through p, so $d_T(\ell, C) = d_T(p, C) + 1$, and $\mathrm{ecc}_T(\ell) = \mathrm{ecc}_T(p) + 1$ (every path from ℓ passes through p, so $d_T(\ell, u) = d_T(p, u) + 1$ for all $u \neq \ell$). Combining with the formula for p:

$$\mathrm{ecc}_T(\ell) = \mathrm{ecc}_T(p) + 1 = \mathrm{rad}(T) + d_T(p, C) + 1 = \mathrm{rad}(T) + d_T(\ell, C). \qquad \square$$

Lemma B.8. *Let T be a finite tree with center C. For every vertex $v \notin C$, let p be the neighbor of v on the (unique) path from v to C. Then*

$$\mathrm{ecc}(p) = \mathrm{ecc}(v) - 1$$

Proof. By Lemma B.7, $\mathrm{ecc}(x) = \mathrm{rad}(T) + d(x, C)$ for every $x \in V(T)$. Since p lies on the path from v to C, $d(p, C) = d(v, C) - 1$, hence $\mathrm{ecc}(p) = \mathrm{ecc}(v) - 1$. $\square$

B.3 CWL IDENTIFIES ALL EMBEDDED TREES

The main result of this section is that the CWL algorithm is a complete invariant for embedded trees:

Theorem 4.2 (CWL identifies all embedded trees)**.** *Let (T, π) and (G, π') be an embedded tree and an embedded graph, respectively, then*

$$(T, \pi) \equiv_{CWL} (G, \pi') \iff (T, \pi) \cong_R (G, \pi').$$

While the backward implication follows directly from Lemma B.3, the forward implication requires more work. We outline the argument here.

- The key structural fact, established in Proposition B.10, is that, in a tree rooted at its center, along any path of length 2 of the form *parent–child–grandchild* the parent and the grandchild receive distinct WL (and hence CWL) colors. This relies on the following results:
  - Lemmas B.7 and B.8 show that the eccentricity strictly decreases along any path from a vertex toward the center.
  - Proposition B.9 shows that the eccentricity of a vertex is determined by its stable WL color. In particular, two vertices with different eccentricities have different stable colors.
- With this distinguishability in hand, we can prove the case in which both T and G are trees: Proposition B.11 constructs, recursively from the center outward, an isomorphism preserving both the CWL colors of the vertices and the rotation system: at each step, the parent's color is unique among the neighbors, which anchors the cyclic order of the children and converts it into a canonical linear order.
- A simple counting argument, given in Lemma B.12, shows that, if $(T, \pi) \equiv_{\text{CWL}} (G, \pi')$, then G must be a tree, concluding the proof.

Proposition B.9. *Let F be a forest and let c denote its stable WL coloring. Distances and eccentricities are computed within the component of the vertex in question. For all $u, w \in V(F)$,*

$$c_u = c_w \implies \mathrm{ecc}(u) = \mathrm{ecc}(w).$$

Proof. By stability of c, any two vertices x, x' with $c_x = c_{x'}$ have the same multiset of neighbor colors. Hence, the following quantities depend only on the color δ, not on the choice of representative x with $c_x = \delta$:

$$\mu(\delta, \gamma) := |\{y \in N(x) : c_y = \gamma\}|, \qquad \deg(\delta) := \sum_{\gamma} \mu(\delta, \gamma).$$

For $v \in V(F)$ and $r \geq 0$, define the following quantity:

$$D_v(\gamma, r) := |\{x \in V(F) : d(v, x) = r \text{ and } c_x = \gamma\}|$$

i.e., the number of vertices at distance r from v of color γ.

Since $\mathrm{ecc}(v) = \max\{r \geq 0 : \sum_{\gamma} D_v(\gamma, r) > 0\}$, it suffices to prove the following.

Claim. $D_v(\gamma, r)$ depends only on c_v, γ, and r.

We argue by strong induction on r. Throughout, $\mathbb{1}[P]$ denotes the Iverson bracket (1 if P holds, 0 otherwise).

Base cases.

$$D_v(\gamma, 0) = \mathbb{1}[c_v = \gamma], \qquad D_v(\gamma, 1) = \mu(c_v, \gamma).$$

Both depend only on c_v and γ.

Inductive step. Fix $r \geq 2$ and assume the claim for all $r' < r$. Root the component of v at v. Every vertex x with $d(v, x) = r$ has a unique parent $\mathrm{par}(x)$ with $d(v, \mathrm{par}(x)) = r - 1$. For each vertex p at distance $r - 1$ from v, the children of p are exactly $N(p) \setminus \{\mathrm{par}(p)\}$, so the number of color-γ children of p is

$$\mu(c_p, \gamma) - \mathbb{1}[c_{\mathrm{par}(p)} = \gamma].$$

Summing over all p at distance $r-1$ from v:

$$D_v(\gamma, r) = \underbrace{\sum_{\delta} D_v(\delta, r-1)\,\mu(\delta, \gamma)}_{(A)} \; - \; \underbrace{\sum_{\substack{p \in V(F) \\ d(v,p)=r-1}} \mathbb{1}[c_{\mathrm{par}(p)} = \gamma]}_{(B)}.$$

Term (A) depends only on c_v, γ, and r by the inductive hypothesis applied at depth $r-1$ together with the WL invariant μ. For term (B), group the sum by $q := \mathrm{par}(p)$. Each vertex q at distance $r-2$ from v is the parent of $\deg(c_q) - \mathbb{1}[d(v,q) > 0]$ vertices at distance $r-1$, where the bracket accounts for q's own parent (absent only when $q = v$, i.e., when $r = 2$). Therefore:

$$(B) = \begin{cases} \deg(c_v) \cdot \mathbb{1}[c_v = \gamma] & \text{if } r = 2, \\ D_v(\gamma, r-2) \cdot (\deg(\gamma) - 1) & \text{if } r \geq 3. \end{cases}$$

In both cases (B) depends only on c_v, γ, r, and the WL invariant $\deg$, using the inductive hypothesis at depth $r-2$ when $r \geq 3$. Hence $D_v(\gamma, r)$ depends only on c_v, γ, and r, completing the induction.

Conclusion. If $c_u = c_w$, then $D_u(\gamma, r) = D_w(\gamma, r)$ for all γ and r. Taking the maximum r with nonzero total yields $\mathrm{ecc}(u) = \mathrm{ecc}(w)$. □

Proposition B.10. *Let T be a component of a forest F, rooted at its center C, and let c be the stable WL coloring of F. For every $v \notin C$, the parent of v and each child of v have distinct colors. The same holds for the stable CWL coloring, which refines c.*

Proof. Since $v \notin C$ and u is a child of v, neither v nor u lies in C. Applying Lemma B.8 to v and u gives $\mathrm{ecc}(p) = \mathrm{ecc}(v) - 1$ and $\mathrm{ecc}(v) = \mathrm{ecc}(u) - 1$, so

$$\mathrm{ecc}(u) > \mathrm{ecc}(v) > \mathrm{ecc}(p).$$

By Proposition B.9, $c_u = c_p$ would force $\mathrm{ecc}(u) = \mathrm{ecc}(p)$, a contradiction. □

Proposition B.11. *Let (T, π) and (T', π') be two embedded trees with centers C and C', both rooted at their respective centers. Let c denote the stable CWL coloring of the disjoint union $(T, \pi) \sqcup (T', \pi')$.*

$$\textit{If} \quad \mathit{CWL}(T, \pi) = \mathit{CWL}(T', \pi') \quad \textit{then} \quad (T, \pi) \cong_R (T', \pi'),$$

i.e., there exists a graph isomorphism $\varphi \colon T \to T'$ that commutes with π and π' and preserves the CWL coloring ($c_v = c_{\varphi(v)}$ for every $v \in V(T)$).

Proof. We first observe that the assumption $\mathrm{CWL}(T, \pi) = \mathrm{CWL}(T', \pi')$ implies the existence of a bijection $\psi \colon C \to C'$ with $c_x = c_{\psi(x)}$ for all $x \in C$. Indeed, $\mathrm{CWL}(T, \pi) = \mathrm{CWL}(T', \pi')$ means that the multisets of stable colors coincide. By Lemma B.1 and Proposition B.9, eccentricity is determined by the stable CWL color, so the sub-multiset of colors realizing the minimum eccentricity is also determined. Hence $\{\{c_x : x \in C\}\} = \{\{c_{x'} : x' \in C'\}\}$, and any color-matching bijection ψ exists.

We construct φ recursively by depth, where the depth of a vertex is its distance to C (resp. C').

Base case (depth 0). Set $\varphi|_C := \psi$. We then specify φ on the children of each $x \in C$. Fix $x \in C$ and write $x' := \psi(x)$.

Case 1: $|C| = 1$. Then $C = \{x\}$, the children of x are all of $N(x)$, and similarly for x'. List them as $u_1, \ldots, u_k$ in the cyclic order π_x (starting from any reference) and $u'_1, \ldots, u'_{k'}$ in the cyclic order $\pi'_{x'}$. By stability of c and $c_x = c_{x'}$,

$$\big(c_{u_1}, \ldots, c_{u_k}\big)_{\sim} = \big(c_{u'_1}, \ldots, c_{u'_{k'}}\big)_{\sim}.$$

In particular, $k = k'$. Pick any cyclic shift $\rho \in \mathbb{Z}_k$ with $c_{u_i} = c_{u'_{\rho(i)}}$ for all i, and define $\varphi(u_i) := u'_{\rho(i)}$.

Case 2: $|C| = 2$. Write $C = \{x, y\}$ and $C' = \{x', y'\}$ with $y' = \psi(y)$. The children of x are $N(x) \setminus \{y\}$, and the children of x' are $N(x') \setminus \{y'\}$. Since π_x is a cyclic permutation of $N(x)$ containing y, it admits a unique linear representative starting at y. Write it as

$$\pi_x : y \mapsto u_1 \mapsto \cdots \mapsto u_k \mapsto y,$$

and analogously

$$\pi'_{x'} : y' \mapsto u'_1 \mapsto \cdots \mapsto u'_{k'} \mapsto y'.$$

By stability of c and $c_x = c_{x'}$,

$$\big(c_y, c_{u_1}, \ldots, c_{u_k}\big)_\sim = \big(c_{y'}, c_{u'_1}, \ldots, c_{u'_{k'}}\big)_\sim, \tag{1}$$

so in particular $k = k'$. Each u_i is at depth 1, while y is at depth 0. By Lemmas B.1 and B.8 and Proposition B.9, $c_y \neq c_{u_i}$ for every i and $c_{y'} \neq c_{u'_j}$ for every j. Hence $c_y = c_{y'}$ appears exactly once in each tuple of (1), so the two tuples are equal as linear tuples:

$$\big(c_y, c_{u_1}, \ldots, c_{u_k}\big) = \big(c_{y'}, c_{u'_1}, \ldots, c_{u'_k}\big).$$

In particular $c_{u_i} = c_{u'_i}$ for $i = 1, \ldots, k$. Define $\varphi(u_i) \coloneqq u'_i$. Proceed analogously at y, anchoring on x.

In both cases, by construction φ commutes with π_x and $\pi'_{x'}$ at each $x \in C$.

Inductive step ($d \geq 1$). Assume φ is defined on every vertex of T at depth $\leq d$, that it is color-preserving, that it maps depth-j vertices of T bijectively onto depth-j vertices of T' for every $j \leq d$, that it preserves the parent relation, and that it commutes with the rotation systems at every vertex of depth $\leq d - 1$.

Fix $v \in V(T)$ at depth $d \geq 1$ and set $v' \coloneqq \varphi(v)$. Since $v \notin C$, $\mathrm{par}(v) \in N(v)$ is well-defined and has depth $d - 1$. The cyclic permutation π_v admits a unique linear representative starting at $\mathrm{par}(v)$. Write it as

$$\pi_v : \mathrm{par}(v) \mapsto u_1 \mapsto \cdots \mapsto u_k \mapsto \mathrm{par}(v),$$

so $(u_1, \ldots, u_k)$ is the linear order of the children of v. Analogously write

$$\pi'_{v'} : \mathrm{par}(v') \mapsto u'_1 \mapsto \cdots \mapsto u'_{k'} \mapsto \mathrm{par}(v'),$$

where $\mathrm{par}(v') = \varphi(\mathrm{par}(v))$ by the inductive hypothesis.

By stability of c and $c_v = c_{v'}$,

$$\big(c_{\mathrm{par}(v)}, c_{u_1}, \ldots, c_{u_k}\big)_\sim = \big(c_{\mathrm{par}(v')}, c_{u'_1}, \ldots, c_{u'_{k'}}\big)_\sim, \tag{2}$$

so $k = k'$. Set $\delta_p \coloneqq c_{\mathrm{par}(v)} = c_{\mathrm{par}(v')}$. Since $v \notin C$, Proposition B.10 applies to the path $u_i - v - \mathrm{par}(v)$: $\delta_p \neq c_{u_i}$ for every i (and similarly $\delta_p \neq c_{u'_j}$ for every j). Hence δ_p appears exactly once in each tuple of (2), so the two tuples are equal as linear tuples:

$$\big(c_{\mathrm{par}(v)}, c_{u_1}, \ldots, c_{u_k}\big) = \big(c_{\mathrm{par}(v')}, c_{u'_1}, \ldots, c_{u'_k}\big).$$

In particular $c_{u_i} = c_{u'_i}$ for $i = 1, \ldots, k$. Define $\varphi(u_i) \coloneqq u'_i$.

This extends φ to depth $d + 1$ and preserves colors and the parent relation. If $k = 0$, both cyclic permutations fix the already matched parents, so the commuting condition holds automatically. If $k \geq 1$, it follows from:

$$\varphi\big(\pi_v(\mathrm{par}(v))\big) = \varphi(u_1) = u'_1 = \pi'_{v'}(\mathrm{par}(v')) = \pi'_{\varphi(v)}(\varphi(\mathrm{par}(v))),$$

for $i < k$,

$$\varphi\big(\pi_v(u_i)\big) = \varphi(u_{i+1}) = u'_{i+1} = \pi'_{v'}(u'_i) = \pi'_{\varphi(v)}(\varphi(u_i)),$$

and

$$\varphi\big(\pi_v(u_k)\big) = \varphi(\mathrm{par}(v)) = \mathrm{par}(v') = \pi'_{v'}(u'_k) = \pi'_{\varphi(v)}(\varphi(u_k)).$$

Conclusion. By induction, φ is defined on all of $V(T)$, is a color-preserving bijection $V(T) \to V(T')$, and commutes with the rotation systems at every vertex. Every edge of T is either an edge between two vertices of C (only when $|C| = 2$, namely the central edge, which is mapped to the central edge of T' via ψ) or has the form $\{u, \mathrm{par}(u)\}$ and is mapped to $\{\varphi(u), \mathrm{par}(\varphi(u))\}$, an edge of T'. Conversely, every edge of T' arises this way. Hence φ is a graph isomorphism preserving the rotation system. □

Lemma B.12. *Let (T, π) be an embedded tree and let (G, π') be any connected embedded graph. Then if CWL$(T, \pi) =$ CWL(G, π'), G is a tree.*

Proof. Every stable CWL color determines the corresponding first-round color, which in turn determines the vertex degree. Hence, T and G have the same number n of vertices and the same degree multiset. Therefore

$$2|E(G)| = \sum_{v \in V(G)} \deg(v) = \sum_{v \in V(T)} \deg(v) = 2(n-1).$$

Thus $|E(G)| = n - 1$. Since G is connected, it is a tree. □

Proof of Theorem 4.2. If $(T, \pi) \cong_R (G, \pi')$, then $\mathrm{CWL}(T, \pi) = \mathrm{CWL}(G, \pi')$ by Lemma B.3. Conversely, if $\mathrm{CWL}(T, \pi) = \mathrm{CWL}(G, \pi')$, then G is a tree by Lemma B.12, and Proposition B.11 gives $(T, \pi) \cong_R (G, \pi')$. □

C Pebble game proofs

C.1 Proof of Lemma 3.4

Lemma 3.4. *Let (G, π) and (G', π') be two embedded graphs, with associated relational structures $\mathcal{R}, \mathcal{D}$ and $\mathcal{R}', \mathcal{D}'$, respectively. Then $(G, \pi) \cong_R (G', \pi') \iff \mathcal{R} \cong \mathcal{R}' \iff \mathcal{D} \cong \mathcal{D}'$.*

Proof. Let $\mathcal{R} = \langle V, E, T \rangle$ and $\mathcal{R}' = \langle V', E', T' \rangle$ be the rotation structures. Suppose first that $\varphi \colon (G, \pi) \to (G', \pi')$ is a rotation-system isomorphism. Since φ is a graph isomorphism, it preserves E, and

$$T(v, u, w) \iff \pi_v(u) = w \iff \pi'_{\varphi(v)}(\varphi(u)) = \varphi(w) \iff T'(\varphi(v), \varphi(u), \varphi(w)).$$

Hence φ is an isomorphism $\mathcal{R} \to \mathcal{R}'$.

Conversely, let $\varphi \colon \mathcal{R} \to \mathcal{R}'$ be an isomorphism. Preservation of E makes φ a graph isomorphism. For $u \in N(v)$, let $w = \pi_v(u)$. Since $T(v, u, w)$, preservation of T gives

$$T'(\varphi(v), \varphi(u), \varphi(w)).$$

By the uniqueness of the successor in the cyclic order at $\varphi(v)$,

$$\pi'_{\varphi(v)}(\varphi(u)) = \varphi(w) = \varphi(\pi_v(u)).$$

Thus φ is a rotation-system isomorphism, proving

$$(G, \pi) \cong_R (G', \pi') \iff \mathcal{R} \cong \mathcal{R}'.$$

Now suppose that $\varphi \colon (G, \pi) \to (G', \pi')$ is a rotation-system isomorphism. Define

$$\widehat{\varphi} \colon D \to D', \qquad \widehat{\varphi}(v, u) \coloneqq (\varphi(v), \varphi(u)).$$

This is a bijection because φ is a graph isomorphism. Moreover,

$$\begin{aligned} \widehat{\varphi}\big((v, u)^\sigma\big) &= \widehat{\varphi}(v, \pi_v(u)) \\ &= (\varphi(v), \varphi(\pi_v(u))) \\ &= (\varphi(v), \pi'_{\varphi(v)}(\varphi(u))) = \widehat{\varphi}(v, u)^{\sigma'}, \end{aligned}$$

and

$$\widehat{\varphi}\big((v, u)^\tau\big) = \widehat{\varphi}(u, v) = (\varphi(u), \varphi(v)) = \widehat{\varphi}(v, u)^{\tau'}.$$

Therefore $\widehat{\varphi}$ preserves R_σ and R_τ, and hence is an isomorphism $\mathcal{D} \to \mathcal{D}'$.

Conversely, let $\Phi \colon \mathcal{D} \to \mathcal{D}'$ be an isomorphism. Since it preserves R_σ and R_τ,

$$\Phi(d^\sigma) = \Phi(d)^{\sigma'} \quad \text{and} \quad \Phi(d^\tau) = \Phi(d)^{\tau'} \qquad \text{for every } d \in D.$$

The σ-orbits are precisely the nonempty stars $D_v = \{(v, u) : u \in N(v)\}$. Hence Φ induces a bijection $\varphi \colon V \to V'$ determined by

$$\Phi(D_v) = D'_{\varphi(v)}.$$

For a dart (v, u), write $\Phi(v, u) = (\varphi(v), x)$. Commutation with τ gives

$$\Phi(u, v) = \Phi(v, u)^{\tau'} = (x, \varphi(v)).$$

Since $\Phi(u, v) \in D'_{\varphi(u)}$, we obtain $x = \varphi(u)$, and therefore

$$\Phi(v, u) = (\varphi(v), \varphi(u)).$$

It follows that φ preserves adjacency. Finally,

$$\begin{aligned}(\varphi(v), \varphi(\pi_v(u))) = \Phi(v, \pi_v(u)) &= \Phi\big((v, u)^{\sigma}\big) \\ &= \Phi(v, u)^{\sigma'} = (\varphi(v), \pi'_{\varphi(v)}(\varphi(u))),\end{aligned}$$

so $\varphi(\pi_v(u)) = \pi'_{\varphi(v)}(\varphi(u))$. Thus φ is a rotation-system isomorphism. This proves

$$(G, \pi) \cong_R (G', \pi') \iff \mathcal{D} \cong \mathcal{D}',$$

and completes the proof. □

C.2 Proof of Theorem 5.1

In this section, we prove Theorem 5.1, which we restate here.

Theorem 5.1. *Let $\mathcal{R} = \langle V, E, T\rangle$ and $\mathcal{R}' = \langle V', E', T'\rangle$ be two non-isomorphic rotation structures and let $m = \min\{|E|, |E'|\}$. Then, Spoiler has a winning strategy in the 4-pebble game $\mathrm{EF}^4(\mathcal{R}, \mathcal{R}')$ using at most $4m + 2$ rounds.*

The proof makes use of the idea of *tracing a word*: Recall that a rotation system is defined by the two maps on darts σ and τ. Any word w on the alphabet $\mathcal{A} = \{\sigma, \sigma^{-1}, \tau\}$ defines a new map $w\colon D \to D$ by concatenation. For a given word w of length k, we denote its letters by $w = w_1 w_2 \cdots w_k$.

For a dart d_0, for which a player has placed a pebble on both endpoints, and a word w, the player can *trace* w starting from dart d_0 using one additional pebble. Let d_0 be the current dart. For $i = 1$ to k, Spoiler does the following depending on the next letter w_i: If $w_i = \tau$ no pebbles are placed, but the new current dart is $d_i := w_i(d_{i-1})$. If $w_i \in \{\sigma, \sigma^{-1}\}$, the dart $d_i := w_i(d_{i-1})$ contains a node v, which is not part of the current dart d_{i-1}. The player places a pebble on v using an unused pebble at the first step or, in any later step, the unique pebble out of the three tracing pebbles that is not placed on d_{i-1}. The dart d_i becomes the new current dart.

We say a pair of darts $(u_1, u_2) \in \mathcal{D}$, $(v_1, v_2) \in \mathcal{D}'$ is *pebbled* if Spoiler placed a pebble on u_1 and u_2 each, and Duplicator placed the corresponding pebbles on v_1 and v_2.

Lemma C.1. *Let d, d' be a pebbled pair of darts. If Spoiler traces a word w in G from d, then Duplicator has to trace w in G' from d' to avoid losing the game.*

Proof. At any point in the game, let $d = (u_1, u_2)$, $d' = (v_1, v_2)$ be the pair of current darts in the tracing of w for Spoiler and Duplicator. Consider a situation where Spoiler places his third pebble on node u_3. Since this corresponds to a letter σ (resp. σ^{-1}) in w, we have $(u_1, u_2, u_3) \in T$ (resp. $(u_1, u_3, u_2) \in T$). Now there exists a unique node v_3 such that $(v_1, v_2, v_3) \in T'$ (resp. $(v_1, v_3, v_2) \in T'$). Since the pebbles are named, Duplicator must place his pebble on v_3, as otherwise, the partial isomorphism would be violated. □

We say a word w is a *loop* at dart d if $d^w := w(d)$ is equal to d. To win the game, Spoiler will select a dart d (by putting pebbles on its endpoints), and for any dart d' that Duplicator may choose as a response, trace a word w, which is a loop at exactly one of d and d'. After tracing the word, Spoiler will be able to force a violation and win the game. The following lemma, restated from Section 5.1, shows the existence of such a word.

Lemma 5.4. *Let $\mathcal{D} = \langle D, R_\sigma, R_\tau\rangle$ and $\mathcal{D}' = \langle D', R'_\sigma, R'_\tau\rangle$ be two non-isomorphic dart structures, let $m = \min\{|D|, |D'|\}/2$, and let $d_0 \in D$ and $d'_0 \in D'$. Then, there exists a word $w \in \mathcal{W}$ of length at most $4m - 1$ such that w is a loop at exactly one of d_0 and d'_0.*

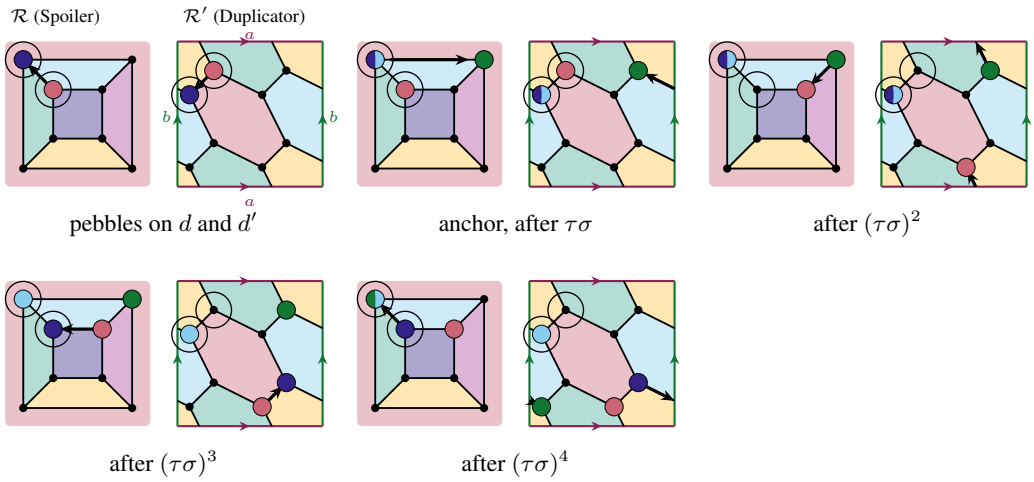

Figure 4: Spoiler's strategy of Theorem 5.1 on the cube of Figure 2c. He traces the face walk $(\tau\sigma)^4$.

Proof. Without loss of generality, assume that $m = |D|/2$. For each $d \in D$ fix a shortest word $r_d \in \mathcal{W}$ with $d_0^{r_d} = d$. We get $|r_d| \leq 2m - 1$ as the underlying graph is connected. Define $\Phi\colon D \to D'$ by $\Phi(d) := (d'_0)^{r_d}$. In particular $\Phi(d_0) = d'_0$. We distinguish three cases.

Case 1: Φ is not injective. Pick $d \neq e$ with $\Phi(d) = \Phi(e)$ and set $w := r_d\, r_e^{-1}$. Then $(d'_0)^w = \Phi(d)^{r_e^{-1}} = \Phi(e)^{r_e^{-1}} = (d'_0)^{r_e\, r_e^{-1}} = d'_0$, while $d_0^w = d^{r_e^{-1}} \neq d_0$, since $d^{r_e^{-1}} = d_0$ would give $d = d_0^{r_e} = e$. Here $|w| \leq 4m - 2$.

Case 2: Φ is injective but does not commute with the action of $\mathcal{W}$. There are $d \in D$ and $s \in \mathcal{A}$ with $\Phi(d^s) \neq \Phi(d)^s$. Set $w := r_d\, s\, r_{d^s}^{-1}$. It is a loop at d_0, since $d_0^w = (d^s)^{r_{d^s}^{-1}} = d_0$. It is not a loop at d'_0:

$$(d'_0)^w = \big(\Phi(d)^s\big)^{r_{d^s}^{-1}} \neq \big(\Phi(d^s)\big)^{r_{d^s}^{-1}} = (d'_0)^{r_{d^s}\, r_{d^s}^{-1}} = d'_0,$$

the middle inequality because $\Phi(d)^s \neq \Phi(d^s)$ and $r_{d^s}^{-1}$ acts injectively. Here $|w| \leq 4m - 1$.

Case 3: Φ is injective and commutes with the action of $\mathcal{W}$, i.e. $\Phi(d^s) = \Phi(d)^s$ for all $d \in D$, $s \in \mathcal{A}$. Then Φ is a bijection: The image of Φ contains d'_0 and is closed under σ and τ. Since the underlying graph is connected, Φ is surjective. Moreover, for $s \in \{\sigma, \tau\}$,

$$R_s(d, e) \Leftrightarrow e = d^s \Leftrightarrow \Phi(e) = \Phi(d)^s \Leftrightarrow R'_s(\Phi(d), \Phi(e)).$$

Hence Φ is an isomorphism between relational structures $\mathcal{D} \cong \mathcal{D}'$. □

Now, we can prove the theorem.

Proof of Theorem 5.1. Let G, π and G', π' be non-isomorphic connected rotation systems. We present a strategy for Spoiler and show that it's a winning strategy.

Spoiler starts by setting the first pebble on an arbitrary node $u_1 \in V(G)$. Duplicator may respond with node $v_1 \in V(G')$. Spoiler sets a second pebble on a node u_2 that is adjacent to u_1. Duplicator has to respond with a node v_2 adjacent to v_1. These four pebbles define the two darts $d := (u_1, u_2)$ in G and $d' := (v_1, v_2)$ in G'.

By Lemma 5.4, there exists a word w, which is a loop at exactly one of d and d'. Without loss of generality, assume it loops at d. Consider the non-equal darts d' and $w(d') =: (v_3, v_4)$. We assume $v_1 \neq v_3$. The case $v_2 \neq v_4$ is proven analogously. Now, Spoiler places a pebble on u_1, which we call the *anchor* and which will never be moved. Duplicator has to respond by placing a pebble on v_1.

Spoiler continues by tracing the word w in G. By Lemma C.1, Duplicator's moves are forced to be tracing w in G'.

Then, as w is a loop in G, the anchor pebble and one of the three pebbles used for tracing by Spoiler lie on the node u_1. However, in G', the anchor lies on v_1, while the tracing pebble arrives at v_3, which are distinct. Hence, after tracing w, the pebbles do not form a partial isomorphism from G to G' and Spoiler wins. As $|w| \leq 4m - 1$ and there are 3 initial moves before w is traced, the total number of rounds is at most $4m + 2$. □

We now generalize the theorem statement to a slightly more general case, where we not only distinguish connected rotation systems, but also rotation systems from any other relational structure of the same signature.

Theorem C.2. *Let $\mathcal{R} = \langle V, E, T\rangle$ be a rotation structure and $\mathcal{R}' = \langle V', E', T'\rangle$ be a non-isomorphic relational structure of the same signature, and let $m = \min\{|E|, |E'|\}$. Then Spoiler has a winning strategy in the 4-pebble game $\mathrm{EF}^4(\mathcal{R}, \mathcal{R}')$ using at most $4m + 2$ rounds.*

Proof. We only need to consider the case where $\mathcal{R}'$ is not a rotation structure or G' is disconnected, as all other cases are covered by Theorem 5.1. We distinguish four cases:

(1) There exist x', y', z' with $T'(x', y', z')$ and $y' \notin N(x')$ or $z' \notin N(x')$: Spoiler pebbles, x', y', z'. No matter where Duplicator chooses to place the corresponding 3 pebbles, either the adjacency or the rotation relation is violated, and Spoiler wins.

(2) T' is not a permutation of the neighbors of a node x', i.e. $T'(x', y', \cdot)$ or $T'(x', \cdot, y')$ does not have a unique solution for $y' \in N_G(x')$: We only consider the former case, as the latter is proven analogously. If both $T'(x', y', z'_1)$ and $T'(x', y', z'_2)$ hold for some x', y', z'_1, z'_2, then Spoiler pebbles these 4 vertices and wins as $T(x, y, \cdot)$ has a unique solution. If $T'(x', y', \cdot)$ does not have any solution, then Spoiler pebbles x' and y'. After Duplicator responds with x and y, Spoiler pebbles the unique successor z of y around x, for which $T(x, y, z)$ holds, and Duplicator cannot respond without violating the partial isomorphism for T'.

(3) T' is a permutation but not a single cycle of the neighbors of a node x'. First suppose $|E| \leq |E'|$. Let $T'_{x'}$ denote the permutation at x'. Spoiler pebbles x'. Let x be Duplicator's answer. Spoiler pebbles $y'_1, y'_2 \in N_{G'}(x')$ in distinct $T'_{x'}$-cycles. Duplicator is forced to answer $y_1, y_2 \in N_G(x)$. As T is a cyclic permutation of $N_G(x)$, there is r such that the word σ^r satisfies $(x, y_2) = \sigma^r((x, y_1))$. Then, Spoiler traces σ^r and wins the game: Duplicator's moves are forced by Lemma C.1 and Duplicator will have 2 pebbles on y_2, while Spoiler can only have one on y'_2. If instead $|E'| \leq |E|$, then the word σ^r might have length exceeding $4m$. In that case, there is an $r' < r$ such that $\sigma^{r'}$ loops at y'_2, but not at y_2. Thus, Spoiler wins after tracing $\sigma^{r'}$. Since $r' \leq |E'| = m$, Spoiler wins in at most $4m + 2$ rounds.

(4) G' is disconnected. Recall that G is connected. First suppose $|E| \leq |E'|$. Spoiler pebbles two vertices u', v' of V' which are not connected. Let u, v be Duplicator's responses. Let P be a path connecting u and v in G. Using the 2 remaining pebbles and the one on u', Spoiler places pebbles moving along P and wins, as Duplicator's moves are forced to preserve the edge relation, and Duplicator will have 2 pebbles on v, while Spoiler can only have one on v'. If $|E'| \leq |E|$, then choose a connected subgraph H' of G'. By Theorem 5.1, Spoiler wins in $4\min\{|E|, |E(H)|\} + 2 \leq 4m + 2$ moves. □

C.3 Proof of Proposition 5.3

Our final result in this section is an example showing that Theorems C.2 and 5.1 are tight in the sense that reducing the k-TWL$^{\mathcal{R}}$ dimension to $k = 2$, i.e., the number of pebbles to three, does not yield completeness.

Proposition 5.3. *2-TWL$^{\mathcal{R}}$ cannot distinguish the planar and the toroidal embeddings of the cube in Figure 2c.*

Proof. Consider the two cellular embeddings of the cube Q_3 in Figure 2c. The planar embedding has six faces of degree 4, whereas the toroidal embedding has four faces, all of degree 6. Hence, they are non-equivalent. Nevertheless, a direct evaluation of 2-TWL$^{\mathcal{R}}$ shows that, in both rotation systems, the stable color of an ordered pair is determined solely by its graph distance in Q_3, with

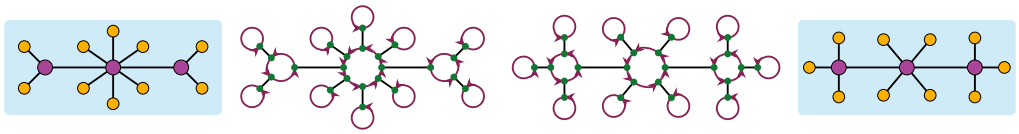

Figure 5: Two abstractly non-isomorphic embedded trees and their corresponding dart representations that 1-TWL$^{\mathcal{D}}$ fails to distinguish.

multiplicities $8, 24, 24, 8$ for distances $0, 1, 2, 3$, respectively. Thus, 2-TWL$^{\mathcal{R}}$ does not distinguish them, and consequently neither does count-free 2-TWL$^{\mathcal{R}}$. □

C.4 Proof of Theorem 5.5

The proof of Theorem 5.5 follows the proof of Theorem 5.1 closely. Instead of pebbling the two endpoints of a dart, Spoiler and Duplicator place pebbles on the darts themselves. The definition of *tracing* is adapted from Appendix C.2 for dart structures in the following way: A player can trace a word w of length k starting from a pebbled dart d_0 using one additional pebble. For $i = 1$ to k, the player pebbles the dart $d_i = w_i(d_{i-1})$ using the pebble that is not placed on d_{i-1}.

Proof of Theorem 5.5. The following describes Spoiler's strategy and proves that it's a winning strategy. Spoiler starts by placing the first pebble on an arbitrary dart $d \in D$. Let d' be Duplicator's response. Spoiler puts another pebble on d, which we call anchor, and Duplicator has to respond by pebbling d'. Now, Spoiler traces the word w given by Lemma 5.4. Analogously to Lemma C.1, Duplicator has to respond by tracing w. As w is a loop in $\mathcal{D}$, but not in $\mathcal{D}'$, Spoiler has two pebbles on d after tracing w, while Duplicator only has one pebble on d', thus violating the partial isomorphism. The number of rounds is bounded by two initial rounds for pebbling d twice and the length of w, which is at most $4m - 1$. □

D Other Expressivity Results

In this section, we present expressivity relationships across different methods. Note that, since 3-TWL$^{\mathcal{R}}$ and 2-TWL$^{\mathcal{D}}$ are complete, when comparing another method against them, it suffices to provide a pair of non-equivalent embedded graphs that such a method fails to distinguish.

Proposition D.1. *CWL $\preceq$ 2-TWL$^{\mathcal{R}}$: for any embedded graphs (G, π) and (G', π') with rotation structures $\mathcal{R}, \mathcal{R}'$, $\mathcal{R} \equiv_{\text{2-TWL}^{\mathcal{R}}} \mathcal{R}'$ implies $(G, \pi) \equiv_{\text{CWL}} (G', \pi')$.*

Proof. Let c_{uv} denote the pair colors after joint stabilization, using shared hashes, and put $b_u = c_{uu}$. Stability means that the partition of the pairs no longer changes, so c_{uv} determines the aggregate $\{\{(\text{atp}_{\mathcal{R}}(u, v, w), c_{wv}, c_{uw}) : w \in V\}\}$ that produced the next color. The substitutions $w = u, v$, identifiable through the equality atoms, show that c_{uv} determines b_u and b_v. For $uv \in E(G)$, the unique substitution satisfying $T(u, v, w)$, namely $w = \pi_u(v)$, similarly shows that c_{uv} determines $c_{u,\pi_u(v)}$.

Suppose $b_u = b_{u'}$. Their diagonal aggregates give equal degrees d and equal multisets of outgoing neighbor-pair colors, the neighbors being identifiable through the edge atoms. Choose $v \in N(u)$ and $v' \in N(u')$ with $c_{uv} = c_{u'v'}$. Iterating the preceding successor property gives $c_{u,\pi_u^i(v)} = c_{u',(\pi'_{u'})^i(v')}$ for $0 \le i < d$. Since pair colors determine endpoint diagonal colors, the cyclic tuples of b-colors around u and u' coincide. Induction on the CWL iterations therefore shows that b refines the stable CWL coloring.

Equality atoms distinguish diagonal from off-diagonal pairs, so equal pair-color multisets imply equal diagonal-color multisets, and hence equal CWL readouts. □

Proposition D.2. *1-TWL$^{\mathcal{D}}$ $\preceq$ 2-TWL$^{\mathcal{R}}$: for two embedded graphs with rotation structures $\mathcal{R}, \mathcal{R}'$ and dart structures $\mathcal{D}, \mathcal{D}'$, $\mathcal{R} \equiv_{\text{2-TWL}^{\mathcal{R}}} \mathcal{R}'$ implies $\mathcal{D} \equiv_{\text{1-TWL}^{\mathcal{D}}} \mathcal{D}'$.*

Proof. Assume $\mathcal{R} \equiv_{2\text{-TWL}^{\mathcal{R}}} \mathcal{R}'$. Let c_{uv} denote their jointly stable pair colors, and write $c_d = c_{uv}$ for $d = (u, v)$. Adjacency atoms identify darts, so the multiset $M = \{\{c_d : d \in D\}\}$ agrees in both structures.

Equality substitutions and diagonal aggregates show that c_d determines $\deg(u)$. The unique substitutions satisfying $T(u, v, w)$ and $T(u, w, v)$ show that it also determines c_{d^σ} and $c_{d^{\sigma^{-1}}}$. Moreover, coordinate reversal preserves color equality (by induction on refinement), so c_d determines c_{d^τ}. Thus c_d determines the colors and coincidences within $S_d = \{d, d^\sigma, d^{\sigma^{-1}}, d^\tau\}$, whose coincidences depend only on whether $\deg(u) = 1$, 2, or at least 3.

The unary atomic type of d records only whether $\deg(u) = 1$. The atomic type of (d, e) is determined by the unary types and which equalities $e = d, d^\sigma, d^{\sigma^{-1}}, d^\tau$ hold. Hence c_d and M determine $\{\{(\mathrm{atp}_{\mathcal{D}}(d, e), c_e) : e \in D\}\}$: the contributions outside S_d follow by subtracting those of its distinct elements from M. Induction on refinement therefore shows that c_d determines every 1-TWL$^{\mathcal{D}}$ color. Equality of M yields equal dart readouts. □

The following proposition completes our expressivity analysis, stating the incompleteness of 2-TWL$^{\mathcal{R}}$, adding strictness to the previous two results and showing the incomparability of CWL and 1-TWL$^{\mathcal{D}}$.

Proposition D.3. *The following hold:*

1. *The two planar embeddings in Figure 2b are distinguished by 2-TWL$^{\mathcal{R}}$ and 1-TWL$^{\mathcal{D}}$, but not by CWL.*
2. *The two embedded trees in Figure 2a are distinguished by 2-TWL$^{\mathcal{R}}$ and CWL, but not by 1-TWL$^{\mathcal{D}}$ (their dart structures are shown in Figure 5).*

Consequently, CWL $\prec$ 2-TWL$^{\mathcal{R}}$, 1-TWL$^{\mathcal{D}}$ $\prec$ 2-TWL$^{\mathcal{R}}$, and CWL $\not\succeq$ 1-TWL$^{\mathcal{D}}$.

Proof. A direct computation proves the proposition. □

E Details about our GNNs

In this section, we provide details on the GNN architectures we propose, on how inputs are encoded, and on our ablation setting.

Inputs and ablations. In general, the input is a graph with a rotation system, π_v for each vertex v. We encode the rotation system on darts: each edge uv gives two darts, (u, v) and (v, u). The permutation τ swaps the two darts of each edge, and $\sigma(v, u) = (v, \pi_v(u))$ moves to the next dart around v. For vertex features, we consider degree values for both the synthetic and MANTRA tasks, and atom and bond descriptors for molecular tasks. We consider an ablation scenario, labelled *shuffled*, where rotation systems are replaced by a random order of the neighbors of a node, while the graph, the features, the architecture, and the parameters remain the same. For molecule encodings, mirror images are encoded from reflected coordinates, not by reversing the cyclic orders (Appendix H).

CGN. The cyclic graph network replaces GIN's neighbor sum with ChiENN-style cyclic aggregation (Xu et al., 2019; Gainski et al., 2023). For each vertex v and neighbor $u \in N(v)$, take the cyclic window starting at u, with $u_i = \pi_v^i(u)$ for $0 \le i < \min\{k, \deg(v)\}$. In the expression below, any remaining feature slots are zero-padded and masked. Let $[\cdot|\cdot]$ denote concatenation. With bond features e_{uv} when available, the message and update are

$$\psi_{uv}^k = W_1 \,\mathrm{ELU}\big(W_2[x_{u_0}|\cdots|x_{u_{k-1}}|e_{uv}] + b_2\big) + b_1, \qquad x'_v = \mathrm{MLP}\left((1+\varepsilon)x_v + \sum_{u \in N(v)} \psi_{uv}^k\right).$$

Every neighbor starts one window, so the sum is invariant under cyclic shifts. We stack L layers with dropout and pool final vertex states by sum or mean before a two-layer head. CGN with $k = 1$ learns an order-blind message. The GIN baseline sums raw neighbor states.

TGN. The topological graph network augments PPGN's pair states (Maron et al., 2019). Initialize

$$H^0_{uv} = \mathrm{MLP}_0([h_u, h_v, \mathbb{1}[u=v], A_{uv}, e_{uv}]), \qquad H \in \mathbb{R}^{n\times n\times C}.$$

For distinct u, v, w, $t_{uvw} \in \{0,1\}^6$ records the six permutations of $T(x,y,z) \Leftrightarrow \pi_x(y) = z$. Set $t_{uvw} = 0$ otherwise, as repeated-index instances are omitted from the sparse correction. A layer computes

$$\begin{aligned} P_{uvw} &= \phi_L(H_{uw}) \odot \phi_R(H_{wv}), \\ M_{uv} &= \sum_w P_{uvw} + \sum_{w:\, t_{uvw}\neq 0} \left(g([P_{uvw}, t_{uvw}]) - g([P_{uvw}, \mathbf{0}])\right), \\ H'_{uv} &= H_{uv} + U([H_{uv}, M_{uv}]). \end{aligned}$$

Here ϕ_L, ϕ_R, U are two-layer ReLU networks. g is a shared one-hidden-layer GELU network with zero-initialized output matrix. The centered correction vanishes for $t = 0$ and at initialization. Only $O(m)$ triples have nonzero type, allowing a precomputed gather/scatter correction alongside the $O(n^3C)$ channelwise matrix product.

The readout concatenates diagonal and off-diagonal sums, divided by n and n^2 respectively for mean pooling, before a two-layer head. Disabling the correction gives our PPGN backbone. Shuffling rebuilds the triples. Synthetic and MANTRA tasks use the plain head. Molecular readouts are defined once in Appendix H.

Implementation. Models use PyTorch (Paszke et al., 2019), PyTorch Geometric (Fey & Lenssen, 2019), Adam, gradient clipping at 1, and best-validation checkpoint selection unless a benchmark protocol specifies otherwise. Schedules, search spaces, selected settings and seed counts are reported below. Reported timings use one NVIDIA RTX 3080 GPU. Configurations originally trained elsewhere were re-timed on that GPU.

F Synthetic benchmarks

In this section, we provide additional detail on the synthetic benchmark. First, we describe how the data is generated and the tasks we consider. Then, we discuss the results of our experiments.

F.1 Data and tasks

We generate 230 graphs with 5 to 20 vertices as follows: 31 named graphs (circulants, generalized Petersen graphs, complete and complete bipartite graphs, prisms, Möbius ladders and cubes), 120 planar graphs using `plantri` (Brinkmann & McKay, 2007) with 8 to 10 vertices, 39 torus triangulations using `surftri` (Sulanke, 2006) with 8 or 9 vertices, and 40 random connected graphs with 8 to 12 vertices and minimum degree 3, sampled as a random spanning tree plus random extra edges. We sample 120 distinct rotation systems for each graph and thus generate a total of 20 922 embeddings. We now describe the three tasks we consider on these embeddings.

Embedding identification. Among the 20 922 embeddings, we identify 17 761 pairs of embeddings of the same graphs that are not isomorphic but that have the same number and size of faces, as well as the same genus. These pairs cannot be simply distinguished by their abstract graph, genus, or face sizes, and thus require using information on the rotation system. We divide the pairs in three groups, based on the weakest separating test that correctly identifies them: 15 107 by CWL, 2651 by 2-TWL$^{\mathcal{R}}$, and three only by methods that are complete on embedded graphs. The last group consists of mirror pairs of equivelar K_7 and $C_{10}(1,3)$ embeddings and two reflexible $K_{3,4}$ embeddings. For each of the groups, we select two pairs of embeddings for each graph: we consider these embeddings as classification classes for the graph. For each class, we generate 10 copies of the embedding with random vertex relabelings. We split the resulting dataset in training, validation, and test instances for each class, with 60/20/20 ratios. Therefore, training and test contain the same graph classes under different relabelings. In summary, the groups contain respectively 715, 114, and 6 classes over 197, 33, and 3 graphs, with 1430, 228, and 12 test instances. A rotation-system oblivious model can identify the abstract graphs, but not distinguish their embedding classes, and is thus expected to obtain a baseline accuracy of $197/715 = 27.6\%$ and $33/114 = 28.9\%$ on the CWL and 2-TWL$^{\mathcal{R}}$ groups, and of 50% on the complete group.

Table 4: Results on the synthetic benchmarks. We report the number of model parameters for each model and average test accuracy (%) ± standard deviation over three seeds. For the identification tasks, the standard deviation is within one percentage point, and we omit it for ease of reading. Graph-only ceilings: 27.6, 28.9 and 50.0 for the identification task, 42.8 for the genus task, and 50.0 for the trees task. Our models are highlighted in purple.

		Identification				
Model	Params	$\mathcal{G}_{\text{CWL}}$	$\mathcal{G}_{\text{2-TWL}^{\mathcal{R}}}$	$\mathcal{G}_{\text{3-TWL}^{\mathcal{R}}}$	Genus	Trees
GIN	142 847	27.6	11.4	50.0	41.6±0.5	50.0±0.0
CGN $k = 1$	132 773	27.5	11.4	50.0	41.6±0.5	50.0±0.0
CGN $k = 3$, shuffled	127 055	27.1	11.1	50.0	41.3±0.1	49.0±0.5
PPGN	226 507	26.3	19.3	50.0	42.1±0.6	50.0±0.0
TGN, shuffled	219 339	26.2	30.1	47.2	42.5±0.2	51.7±0.3
CGN $k = 3$	127 055	100.0	11.1	50.0	52.0±0.8	96.3±1.2
TGN	219 339	100.0	97.7	50.0	92.8±1.5	100.0±0.0

Genus on unseen graphs. From the 20 922 embeddings, we generate two random vertex relabelings for each and label each resulting in 41 844 instances. In this case, we split by graphs: the 230 graphs are divided 60/20/20 into training, validation, and test graphs, resulting in 25 208 training, 8126 test, and 8510 validation instances. Note that in this experiment the 46 test graphs are not seen by the training procedure. The task is to predict the genus of the embedding on unseen graphs. As a baseline, predicting the most frequent genus yields 38.1% accuracy. A model that is oblivious to the rotation system is expected to achieve at best the accuracy of predicting the most frequent test genus for each graph, resulting in an accuracy of 42.8%.

Embedded tree. For this task, we consider 50 random trees on 10 to 40 vertices, each with two embeddings that differ by swapping two neighbors at one internal vertex. As in the identification task, we label each embedding with 10 random relabelings, split again 60/20/20. In this case, a rotation-system-oblivious approach yields at most 50% accuracy.

F.2 Training and results

All the models we consider have 4 layers and use a batch size of 16. We use 10 warmup epochs and cosine decay. CGN ($k = 3$) and TGN have width 64. For the other models, we set the width to roughly match the number of parameters of CGN ($k = 3$) and TGN: GIN uses width 108, CGN with $k = 1$ uses 78, and PPGN uses 72. The GIN and CGN models use a learning rate of $5 \cdot 10^{-4}$, with 300 epochs for the identification and trees tasks, and 150 for genus task. The TGN and PPGN models use a learning rate of $3 \cdot 10^{-4}$ with 800 epochs for the identification task, 400 epochs for genus task, and a learning rate 10^{-3} with 1600 epochs for trees task. The rate of $3 \cdot 10^{-4}$ was selected on the 2-TWL$^{\mathcal{R}}$ validation set from the range 10^{-4} to 10^{-2}. We use three training seeds on one fixed split.

Table 4 presents the results. The CGN and TGN models reproduce the theoretical expressivity of their tests: they both solve the CWL group, only TGN solves the 2-TWL$^{\mathcal{R}}$ group, and neither exceeds chance on the complete group. Shuffling the rotation systems brings each model down to abstract-graph-only performance, and CGN with $k = 1$ behaving as a GIN model. On the 2-TWL$^{\mathcal{R}}$ group, the 19.3% performance of PPGN coincides with the ceiling of identification by abstract 1-WL, which separates 22 of the group's 33 graphs, below its own 2-WL ceiling of 28.9%. TGN with shuffled rotation system reaches 30.1% On the genus task, TGN exceeds the graph-only oracle by 50 points and CGN by 9, whereas shuffling removes both improvements. On the trees task, where CWL is complete (Theorem 4.2), CGN detects most transpositions and mean-pooled TGN reaches 100%.

Computational cost. We measure runtimes and memory on a single NVIDIA RTX 3080 GPU. On the 2-TWL$^{\mathcal{R}}$ group in the identification task, an epoch of TGN takes roughly 2.4× as long as GIN and 2× as long as CGN. On the same task, TGN uses 126 MB of peak memory compared with 21–27 MB for GIN and CGN.

G EXPERIMENTS ON MANTRA

In this section we provide additional detail on the experiments on the topological property prediction dataset MANTRA (Ballester et al., 2025). First, we describe the benchmark and the tasks. Then, we describe how we encode the data. Then, we present the results of our experiments.

G.1 DATA, TASK AND METRICS

We consider MANTRA v0.0.19 (Ballester et al., 2025), a benchmark dataset that contains $43\,148$ triangulations of closed connected surfaces on at most 10 vertices. The data is divided into $60/20/20$ splits, resulting in 8631 test surfaces.

Orientability. We consider the binary task of correctly identifying orientable surfaces which we consider the positive class. The dataset is heavily unbalanced, as only 7.9% of the test instances are orientable. We measure the AUROC and the balanced accuracy, i.e., the AUROC at threshold 0.5.

Homeomorphism type. Multiclass classification task with 11 classes, measured by accuracy. The homeomorphism type is determined by the orientability together with the vertex and edge counts.

Betti numbers. Multiclass classification task with 3 classes, measured by accuracy. The Betti numbers are determined by the orientability together with the vertex and edge counts. For this task, we predict the numbers as real-valued outputs and round them to evaluate them in a classification task.

For 1779 orientability test surfaces (20.6% of the total), the dataset contains both orientable and non-orientable triangulations with an isomorphic 1-skeleton. In these cases, the graph alone cannot determine the label. We call this subset of test instances the *hard subset.*

G.2 ENCODING

We encode each of the triangulations as a graph with a cyclic order on the neighbors of each vertex. The cyclic order is determined by the cycle formed by the edges that are opposite the vertex in its incident triangles, i.e., its vertex link. We use the vertex numbering to fix the direction of each order: we start from the smallest-numbered neighbor and take steps towards the smaller link neighbors. This choice depends only on the numbering, not on the label, and we refer to this as the *canonical representative*. Reversing the directions of the links independently preserves the surface and the labels, while shuffling the cyclic orders destroys the link structure and serves as an ablation to test whether the model relies on this information. Vertex features are encoded as learned degree embeddings; comparisons with published settings use scalar degrees. We refer to this

G.3 TRAINING AND RESULTS

Unless stated otherwise, we consider models with width 64, four layers, and batch size of 128. We consider 5 seeds. CGN uses windows of length $k = 3$. We train for either 6 or 100 epochs, using 5 warmup epochs and cosine decay for the latter case, and learning rate 10^{-3}. We report the performance at the best validation checkpoint.

Results are reported in Table 3a. Ballester et al. (2025) report 65 ± 7 balanced accuracy and $64 \pm 26\%$ accuracy for CellMP using scalar degrees, sum readout, six epochs and learning rate 0.01. Rerunning their released code, we obtain 52 ± 2 balanced accuracy, 84 ± 8 AUROC and $92 \pm 0\%$ accuracy. For comparison, in the same setting, TGN with learning rate 10^{-3}, perfectly solves the tasks. TGN also solves the homeomorphism type and Betti numbers tasks, with perfect accuracy over 5 seeds at 100 epochs.

Reproduction and computational cost. We could not reproduce the CellMP results reported in Ballester et al. (2025). The reported balanced accuracy is 65 ± 7, while rerunning their code gives 52–54. Table 3a reports both results. With respect to runtime, CellMP requires 140 seconds per epoch on our GPU, against 2.8 seconds for TGN.

G.4 Ablation

We report here our ablation results on orientability (Table 5). We use degree embeddings and sum readout, both on the full test split and on the hard subset. Only TGN solves the task, but shuffling the rotation systems nullifies its advantage over the other models, confirming that the rotation system is indeed essential for good performance. Longer training does not close the gap between TGN and the other models.

Table 5: MANTRA orientability ablation. We report balanced accuracy, with AUROC in round brackets. Our models are highlighted in purple.

		6 epochs		100 epochs	
Model	Params	full	hard	full	hard
GIN	37 381	53 (89)	50 (57)	56 (90)	50 (54)
GIN, h=140	172 205	53 (89)	50 (57)	56 (90)	50 (55)
PPGN (no rotation)	138 049	58 (90)	51 (57)	55 (90)	50 (58)
CGN k=3, shuffled	103 429	50 (89)	50 (57)	54 (90)	50 (56)
TGN, shuffled	172 865	55 (89)	50 (57)	57 (90)	50 (56)
TGN, h=30	39 211	100 (100)	100 (100)	100 (100)	100 (100)
CGN k=3	103 429	50 (89)	50 (57)	55 (91)	50 (63)
TGN	172 865	100 (100)	100 (100)	100 (100)	100 (100)

G.5 Robustness to link orientation

In this experiment, we test whether fixing the link directions can unfairly favor the TGN model. A surface is orientable if the link directions can be chosen consistently in all vertices. As our encoding itself fixes these directions, we check whether this information can be used by TGN to achieve better performance. To this end, at test time, we reverse the link direction at a random half of the vertices on each surface, which preserves the surface and its label. We report the results in Table 6.

Table 6: TGN on MANTRA orientability with different link directions in training and testing. Canonical directions are those which are fixed by our encoding from the vertex numbering. Random directions are drawn once per surface (fixed) or newly at each epoch (resampled). We measure the performance over canonical test directions, over reversing the link direction of half of the vertices (reversed), and over replacing the cyclic order at every vertex with a random one (shuffled). For the random training directions, the test directions can be canonical, reversed, or shuffled (any).

Training directions	Test directions	Bal. acc.	AUROC
canonical	canonical	100.0	100.0
canonical	reversed	96.2–97.2	99.4–99.5
canonical	shuffled	54.2	76.9
random, fixed	any	54.2–55.2	89.7–90.0
random, resampled	any	54.2–54.7	89.8–90.0

Despite this operation, TGN retains 99.4–99.5 AUROC over two random draws, whereas shuffling the cyclic orders lowers it to 76.9. Therefore, TGN does rely on the link structure itself, and not on the direction choice of our encoding. More specifically, training on random link directions fails: the models obtained with this training strategy stay at the level of graph-only baselines on every test input. Consistent directions are, thus, required to learn the task, even though a trained model tolerates random re-orientation.

H Experiments on ChIRo

In this section we provide additional detail on the experiments on the ChIRo (Adams et al., 2022) benchmark. First, we describe the dataset and the tasks. Then, we provide details on our model and on how we encode the data. The results are reported in Table 3b.

Dataset. ChIRo (Adams et al., 2022) supplies 39 256 enantiomer pairs for the R/S prediction task and 34 560 for the ranking task. Each molecule has exactly one tetrahedral stereocenter. We use the provided train-test split of the data, which ensures that the underlying graph of all test molecules does not appear in the training data.

Obtaining the rotation system from 3D embeddings. To obtain the rotation system from the given 3D embedding of a molecule, we make use of a canonical ordering of the nodes, namely RDKit's canonical atom ranking (Schneider et al., 2015) computed on the graph without stereochemical information. An atom with at most two neighbors has a unique cyclic order of its neighbors. A three-neighbor atom is oriented by the sign of the triple product $a \cdot (b \times c)$, where a, b, c are the direction vectors to the neighboring atoms. For a node u with $d \geq 4$ neighbors, we choose two reference neighbors v, w based on the canonical ordering of the atoms. Then, we project the 3D coordinates of all neighbors to 2D along the axis defined by u, v, i.e., such that the reference neighbor is projected to $(0, 0)$. The rotation system is now defined as $(v, w, x_1, .., x_{d-2})$, where $x_1, ..., x_{d-2}$ are the non-reference neighbors of u ordered according to their angle to w in the 2D projection.

Details of our model. The architecture of CGN and TGN is explained in Appendix E. We use 41 atom and 7 bond features and windows of size $k = 3$ for CGN. All other labels (chiral tags, CIP codes, bond stereo) are removed from the input features, so the two enantiomers of a pair differ only in their rotation systems.

To account for reflections of the molecule, we run our model on both the rotation system π obtained from the given 3D coordinates and the rotation system π_M obtained from the reflected 3D embedding. Let $\mathbf{h}, \mathbf{h}_M$ be the corresponding pooled graph embeddings and let $\mathbf{h}_{\text{sym}}, \mathbf{h}_{\text{anti}} = \frac{1}{2}(\mathbf{h} \pm \mathbf{h}_M)$. Then, the head is

$$y = g(\mathbf{h}_{\text{sym}}) + \langle w, \mathbf{h}_{\text{anti}} \rangle,$$

whose first term is reflection-invariant and whose second changes sign under reflection. ChIRo uses y with R/S threshold 0.

CGN is trained with batch size 128 and TGN with batch size 64. We use three seeds and a learning rate of 10^{-3}. We use learning-rate warmup followed by cosine decay, training for 30 epochs on R/S classification and 100 epochs on ranking.

For ranking, we sample up to four conformers per stereoisomer during training. At evaluation, predictions are averaged over the conformers of each stereoisomer before comparing the two members of an enantiomer pair and ties count as $\frac{1}{2}$.

Details about ChiENN. ChiENN (Gainski et al., 2023) published an accuracy of $76.0 \pm 0.2\%$ on ChIRo. However, they adapted the dataset by merging all conformers of one enantiomer to one representative conformer, which they regenerated. Furthermore, their model uses additional random walk encodings, which improve the performance of their model at the expense of extra runtime (see Table 3b). In Table 3b, we list both the original ChiENN results as well as their performance on the original dataset and without adding random walk positional encodings to allow for a fair comparison with our methods and previous work.