Beyond Weisfeiler–Lehman: Topological Features Increase Graph Neural Network Expressivity

Luke Jones[1] Lucas Schardt[1] Aaditya Bilakanti[2] Taro Iyadomi[3] William Chang[1]

[1]University of California, Los Angeles [2]Phillips Exeter Academy [3]Northwestern University

Abstract

The expressive power of graph neural networks (GNNs) is fundamentally limited by their connection to the Weisfeiler–Lehman (WL) hierarchy. We introduce a class of topologically-augmented equivariant graph neural networks that combine k-order tensor representations with persistent homology via equivariant filtrations on simplicial complexes. We prove that our architecture is at least as expressive as the k-WL test and can be strictly more expressive by leveraging global topological features. We further show that key implementation choices—including filtration construction, persistence diagram vectorization, and feature fusion—preserve these expressivity guarantees. Our results establish a principled connection between combinatorial and topological methods for graph learning, demonstrating that persistent homology can provably enhance graph representation learning beyond what is achievable by equivariant networks alone.

1 Introduction

Graph learning is a highly active research domain in machine learning, fuelled in large part by the *geometric deep learning* paradigm [8, 9] and the rapid development of neural architectures for structured data. While graph neural networks (GNNs) have achieved strong empirical performance across domains ranging from molecular property prediction to social network analysis, their expressive power is fundamentally limited by their reliance on local message passing. In particular, standard message-passing architectures are known to be at most as expressive as the 1-dimensional Weisfeiler–Lehman (WL) test [51, 41], meaning there exist simple pairs of non-isomorphic graphs that no message-passing GNN can distinguish.

This fundamental limitation has motivated two complementary lines of research. The first pursues higher-order equivariant neural networks, which extend message passing to k-tuple interactions and can provably match the expressive power of k-WL [36, 41]. [36] showed that invariant graph networks operating on k-order tensors are as powerful as k-WL, and further demonstrated that a reduced 2-order model augmented with matrix multiplication achieves 3-WL expressiveness while remaining scalable. The second direction incorporates topological methods, particularly persistent homology, which capture multi-scale structural features such as cycles, cavities, and higher-order connectivity patterns that are invisible to local message passing [27, 13, 30, 44].

Recent work has begun to analyze the expressive power of topological methods in graph learning. [2] provided the first comprehensive characterization of persistent homology's expressivity relative to the WL hierarchy, proving that persistent homology is at least as expressive as the corresponding WL test and can distinguish certain graph pairs that k-WL cannot; their analysis shows that the Vietoris–Rips filtration and the WL hierarchy are *incomparable*, suggesting topological and combinatorial methods are genuinely complementary. On the architectural side, [21] introduced simplicial neural networks that generalize graph convolution to simplicial complexes via the Hodge

Laplacian, but focused on signal processing on fixed complexes rather than on expressivity relative to graph isomorphism tests. Despite this progress, higher-order equivariant networks and topological methods have largely been studied in isolation. This paper bridges that gap by introducing a unified framework that combines higher-order equivariant representations with topological features derived from persistent homology, and provides a joint expressivity analysis that captures both local combinatorial refinement and global topological structure within a single architecture. A more complete description of related works is given in Appendix A.

Contributions. We introduce and analyze a class of *topologically-augmented equivariant graph neural networks* that combine k-order tensor representations with persistent homology. First, we prove the architecture is *at least as expressive as k-WL*: the equivariant component simulates k-WL, recovering [36], while a gated topological branch (setting gate parameters to $-\infty$ recovers the pure equivariant network) ensures the topology-augmented hypothesis class strictly contains the equivariant one. Second, we establish a *strict expressivity gain*: there exist non-isomorphic graphs indistinguishable by k-WL but distinguished by our model, extending [2] (who showed persistent homology alone can separate strongly regular pairs invisible to 2-FWL) by preserving this discriminating power within a learnable end-to-end architecture. Along the way, we formalize *equivariant filtration generators* and prove they induce isomorphism-invariant persistence diagrams, and verify that practical design choices (attention-pooling, gated residual fusion, node-level features) preserve expressivity relative to the idealized model, closing the theory-practice gap noted by [36]. Third, we empirically validate both expressivity claims. On five pairs of cospectral, 3-WL-equivalent strongly regular graphs spanning orders 16 to 49 (drawn from the Rook–Shrikhande, $T(8)$–Chang, and Paley–Latin-square families), our randomly-initialized model produces output differences four to seven orders of magnitude above the baseline floating-point noise, converting the theoretical separation beyond 3-WL into a directly verifiable empirical property. On the TU Dortmund benchmarks [53], persistent homology delivers consistent +2.7% to +3.8% accuracy gains across four molecular datasets (NCI1, NCI109, MUTAG, PTC). A diagnostic analysis of the gate dynamics (deferred to Appendix E) further reveals that the deep-layer gate at convergence acts as a direct readout of filtration informativeness.

2 Preliminaries

Graphs and notation. Let $G = (V, E)$ be a finite undirected graph with $|V| = n$. We denote by S_n the symmetric group acting on V. Two graphs $G = (V, E)$ and $G' = (V', E')$ are *isomorphic*, written $G \cong G'$, if there exists a bijection $\varphi : V \to V'$ such that $(v_1, v_2) \in E$ if and only if $(\varphi(v_1), \varphi(v_2)) \in E'$. The *graph isomorphism problem*, determining whether two given graphs are isomorphic, has no known polynomial-time algorithm in general, though efficient algorithms exist for special graph families [40].

Weisfeiler–Lehman hierarchy. The k-dimensional Weisfeiler–Lehman (k-WL) algorithm iteratively assigns colors to k-tuples $\mathbf{i} = (i_1, \ldots, i_k) \in V^k$. At iteration $t+1$, the color $c_{\mathbf{i}}^{(t+1)}$ depends on the current color and the multisets of colors observed in each of k neighborhoods:

$$c_{\mathbf{i}}^{(t+1)} = \mathrm{HASH}\Big(c_{\mathbf{i}}^{(t)},\ \{\{c_{\mathbf{j}}^{(t)} : \mathbf{j} \in N_1(\mathbf{i})\}\}, \ldots, \{\{c_{\mathbf{j}}^{(t)} : \mathbf{j} \in N_k(\mathbf{i})\}\}\Big),$$

where $N_\ell(\mathbf{i}) = \{(i_1, \ldots, i_{\ell-1}, v, i_{\ell+1}, \ldots, i_k) : v \in V\}$ denotes the ℓ-th neighborhood of the tuple $\mathbf{i}$. Two graphs are declared non-isomorphic if the resulting multisets of colors differ. Several key

relationships are known: 1-WL and 2-WL have equivalent discrimination power; k-FWL (Folklore WL) is equivalent to $(k+1)$-WL for $k \geq 2$; and for each $k \geq 2$ there exist non-isomorphic graphs distinguishable by $(k+1)$-WL but not by k-WL [36]. We refer to [40] for a comprehensive survey.

Tensor representations and equivariance. Following [36, 37, 34], we represent graphs using k-order tensors $X \in \mathbb{R}^{d \times n^k}$. Let $[n] = \{1, \ldots, n\}$. The symmetric group S_n acts on such tensors by permuting indices: $(g \cdot X)_{i,j} = X_{g^{-1}(i),j}$ where $j \in [d]$ and g acts on i element-wise. A map $\mathcal{L} : \mathbb{R}^{d \times n^k} \to \mathbb{R}^{d' \times n^k}$ is *equivariant* if $\mathcal{L}(g \cdot X) = g \cdot \mathcal{L}(X)$ for all $g \in S_n$; it is *invariant* if $\mathcal{L}(g \cdot X) = \mathcal{L}(X)$. [37] characterized the space of all linear equivariant layers, showing that for $k = 2$ the equivariant linear maps can be written as combinations of 15 basis operators corresponding to distinct index equality patterns (identity, transpose, row sums, column sums, global sums, and their interactions). An equivariant graph network is defined recursively by $X^{(\ell+1)} = \sigma(\mathcal{L}_\ell(X^{(\ell)}))$, where each $\mathcal{L}_\ell$ is equivariant and σ is a pointwise nonlinearity.

Simplicial complexes and clique complexes. A *simplicial complex* K over a vertex set V is a collection of subsets $\sigma \subseteq V$ (called *simplices*) that is closed under inclusion: if $\sigma \in K$ and $\tau \subseteq \sigma$, then $\tau \in K$. A simplex σ with $|\sigma| = p + 1$ is called a p-simplex. We consider the *clique complex* $K(G)$ of a graph G, in which the simplices are exactly the cliques of G: vertices are 0-simplices, edges are 1-simplices, triangles are 2-simplices, and so on. The clique complex construction is natural for graph learning because it lifts the pairwise connectivity structure of G to a higher-dimensional topological object while preserving graph isomorphism: $G \cong G'$ if and only if $K(G)$ and $K(G')$ are isomorphic as simplicial complexes. As noted by [21], simplicial complexes provide a natural framework for modelling higher-order interactions that go beyond pairwise relationships.

Filtrations and equivariance. A *filtration* of a simplicial complex K is a nested sequence of subcomplexes $\emptyset \subseteq K^{a_0} \subseteq K^{a_1} \subseteq \cdots \subseteq K^{a_m} = K$, typically induced by a scalar-valued *filtration function* $f : K \to \mathbb{R}$ via sublevel sets $K^a = \{\sigma \in K : f(\sigma) \leq a\}$. Intuitively, f assigns a "birth time" to each simplex, and the filtration reveals how the topological structure of K evolves as we sweep through increasing threshold values. In practice, filtrations arise from *filtration generators* $F : \mathcal{G} \to \mathbb{R}^K$ that assign to each graph G a filtration function f_G on its clique complex $K(G)$. We say F is *equivariant* if for any isomorphism $\varphi : G \to G'$, we have $f_G = f_{G'} \circ \varphi$; that is, the filtration values are consistent under relabelling of vertices.

More generally, following [2], one can consider filtration generators that operate on arbitrary simplicial complexes: a map $F : K \in \mathcal{K} \mapsto f_K \in \mathbb{R}^K$ is equivariant if for each graph isomorphism $\varphi : G \to G'$, the induced map on clique complexes satisfies $f_{K_G} = f_{K_{G'}} \circ \varphi$, preserving both the combinatorial structure and the filtration values. This more general formulation allows filtration values to be assigned not only to vertices and edges but also to higher-dimensional simplices, enabling the detection of higher-order topological features. The equivariance condition is crucial: it guarantees that the resulting persistence diagrams are *invariant* under graph isomorphisms, so that topological features serve as well-defined graph descriptors.

Persistent homology. Given a filtration $\{K^a\}_{a \in \mathbb{R}}$ of a simplicial complex, *persistent homology* tracks the birth and death of topological features—connected components, loops, cavities, and their higher-dimensional analogues—as the threshold a increases. The result is a collection of *persistence diagrams* $D_p = \{(b_i, d_i)\}$ for each homological dimension $p \geq 0$, where each point (b_i, d_i) records the birth time b_i and death time d_i of a p-dimensional feature. The *persistence* $|d_i - b_i|$ measures the prominence of each feature: long-lived features represent robust topological structures, while

short-lived ones are typically noise. We refer to [25] for the algebraic foundations and to [2] for the connection to graph learning.

A central property is that equivariant filtration generators yield isomorphism-invariant persistence diagrams.

Proposition 1 (Invariance of persistence diagrams [2, Proposition 1])**.** *Let F be an equivariant filtration generator and let $G \cong G'$. Then the persistence diagrams of f_{K_G} and $f_{K_{G'}}$ coincide for all homological dimensions p.*

This result guarantees that persistent homology is compatible with equivariant representation learning: any topological features derived from an equivariant filtration are genuine graph invariants. As observed by [2], the computation of persistent homology also carries information about graph-theoretic quantities such as the diameter and girth of a graph, suggesting that persistent homology captures more than "just" topological information.

3 Topologically-Augmented Equivariant Graph Networks

We introduce a hybrid graph neural network architecture that augments k-order equivariant representations with topological features derived from persistent homology. The model preserves permutation equivariance while incorporating global structural information beyond local message passing. Our design is motivated by the complementarity between the WL hierarchy and persistent homology established by [2]: since neither method entirely contains the other in general, combining them within a single architecture should yield strictly greater expressive power.

Let $G = (V, E)$ be a graph with $|V| = n$. The model operates on k-order tensors $X^{(\ell)} \in \mathbb{R}^{d_\ell \times n^k}$, which assign a d_ℓ-dimensional feature vector to each ordered k-tuple of vertices. We focus the exposition on the pairwise case $k = 2$, in which $X^{(\ell)} \in \mathbb{R}^{d_\ell \times n \times n}$; the construction below generalizes to arbitrary $k \geq 2$ by replacing ordered pairs $P(\sigma) \subseteq \sigma \times \sigma$ with ordered k-tuples $P_k(\sigma) \subseteq \sigma^k$ throughout, and we state Theorem 3 in this general form. The implementation in Section 4 specializes to $k = 2$, which already attains 3-WL expressivity via matrix multiplication [36]. Each layer ℓ consists of four stages: an equivariant update, a filtration construction, a persistent homology computation, and a feature fusion step. In the next section we discuss the details of the code pipeline and provide a figure of the filtration in Appendix B.1.

Equivariant update. We first apply a permutation-equivariant transformation

$$X^{(\ell+\frac{1}{2})} = \mathrm{EqvLayer}_\ell(X^{(\ell)}),$$

where $\mathrm{EqvLayer}_\ell$ satisfies $\mathrm{EqvLayer}_\ell(g \cdot X) = g \cdot \mathrm{EqvLayer}_\ell(X)$ for all $g \in S_n$. As shown by [36], such layers correspond to k-order equivariant networks and, when composed with appropriate nonlinearities, can simulate k rounds of the k-WL test. In practice, we parameterize the equivariant layer using parallel MLPs with (generalized) matrix multiplication (see Section 4).

Filtration construction. We construct the clique complex $K(G)$ and define a learnable simplex score on it. At layer ℓ, this score is computed from the post-equivariant tensor $X^{(\ell+\frac{1}{2})}$ produced by the equivariant update above. For each simplex $\sigma \in K(G)$, we set

$$f_\ell(X^{(\ell+\frac{1}{2})}, \sigma) = \rho_\ell\left(\frac{1}{|P(\sigma)|} \sum_{(i,j)\in P(\sigma)} X^{(\ell+\frac{1}{2})}_{:,i,j}\right),$$

where ρ_ℓ is a shared MLP and $P(\sigma) = \sigma \times \sigma$ (replaced by σ^k in the k-order generalization). Since $X^{(\ell+\frac{1}{2})}$ transforms equivariantly under permutations and the aggregation over $P(\sigma)$ is invariant to vertex labelling within σ, the raw scores f_ℓ are equivariant under graph isomorphisms.

The map f_ℓ is not guaranteed to be monotone in the face relation, so we define the actual filtration $\tilde{f}_\ell$ by a non-decreasing correction processed in order of increasing dimension; the explicit recursion and its differentiable implementation are given in (1) of Section 4.1. Both f_ℓ and $\tilde{f}_\ell$ are equivariant in the joint sense formalized below; the max-correction preserves equivariance because it depends only on the partial order of faces.

Lemma 2 (Equivariance of the filtration)**.** *For any permutation $g \in S_n$, simplex $\sigma \subseteq V$, and tensor $X \in \mathbb{R}^{d\times n\times n}$, $\tilde{f}_\ell(g \cdot X,\, g \cdot \sigma) = \tilde{f}_\ell(X, \sigma)$.*

The proof, a short computation, is deferred to Appendix B.2. The corrected filtration induces a nested sequence of subcomplexes $K_\ell^a = \{\sigma \in K(G) \mid \tilde{f}_\ell(X, \sigma) \leq a\}$, from which we compute persistence diagrams $D_p^{(\ell)} = \mathrm{PH}_p(K(G), \tilde{f}_\ell)$ for each homological dimension p. By Proposition 1, these persistence diagrams are invariant under graph isomorphisms.

Topological feature extraction. We map persistence diagrams to vector features via $T^{(\ell)} = \Phi_{\mathrm{PH}}(D_0^{(\ell)}, D_1^{(\ell)}, \dots) \in \mathbb{R}^{d_T}$, where Φ_{PH} is a differentiable vectorization map. In the idealized architecture, Φ_{PH} can be any fixed or learned embedding such as persistence images or landscapes [13]. The vector $T^{(\ell)}$ is a *global* (graph-level) topological descriptor that summarizes the multi-scale structural information captured by persistent homology at layer ℓ.

Feature fusion. The topological features are fused into the pairwise representation by

$$X_{uv}^{(\ell+1)} = \psi_\ell(X_{uv}^{(\ell+\frac{1}{2})} \parallel T^{(\ell)}),$$

where ψ_ℓ is an MLP and $\parallel$ denotes concatenation. The global topological vector $T^{(\ell)}$ is broadcast identically to every pair (u, v), ensuring that the fusion preserves equivariance: since $T^{(\ell)}$ is invariant under permutations and $X^{(\ell+\frac{1}{2})}$ is equivariant, the concatenated input transforms equivariantly, and the pointwise MLP preserves this property. The full network is defined recursively as $X^{(\ell+1)} = \mathcal{F}_\ell(X^{(\ell)})$, with final output $\hat{y} = \mathrm{READOUT}(X^{(L)})$, where READOUT is a permutation-invariant function (e.g., summation or pooling).

3.1 Expressivity Analysis

Our main results establish that the model is at least as expressive as k-WL (Theorem 3) and that the topological branch of the implemented model in the next section provides a strict expressivity gain.

Theorem 3 (Lower bound via k-WL)**.** *Fix $k \geq 2$ and consider the k-order instantiation of the architecture described above (operating on $X \in \mathbb{R}^{d\times n^k}$, with the filtration aggregating over ordered k-tuples $P_k(\sigma) \subseteq \sigma^k$). This architecture is at least as expressive as the k-WL test: if two non-isomorphic graphs G and G' are distinguished by k-WL, then there exist model parameters such that the architecture produces different outputs on G and G'. In particular, the pairwise ($k = 2$) instantiation of Section 4, which augments the equivariant layer with matrix multiplication [36], is at least as expressive as 3-WL.*

The proof shows that the equivariant branch alone can simulate k-WL by encoding the color multisets, and that the topological branch does not reduce expressivity; the proof in its entirety is deferred to Appendix B.3. We will prove in Section 4 (Corollary 6) that the implemented $k = 2$ model is strictly more expressive than 3-WL. Computational considerations (clique complex size, persistent homology cost, GPU–CPU transfers) are analyzed in Appendix G.

4 Implementation

We now describe the code implementation of the architecture from Section 3. The implementation differs from the idealized description in several ways; we describe each below and verify that they preserve the equivariance and invariance properties required by our expressivity results. We defer the discussion of the tasks to which we apply the model to Section 5.

Overall pipeline. The model operates on batches of graphs grouped by node count, each represented as a $d \times M \times M$ tensor with the adjacency matrix on channel 0 and node features on the diagonals of channels $c \geq 1$, following the convention of [36]. Each EqvLayer_ℓ is parameterized as the sum of a linear equivariant map in the basis of [37] and a matrix-multiplication branch $\text{MLP}_1(X) \cdot \text{MLP}_2(X)$ applied pointwise, followed by a nonlinearity. At each layer ℓ, the pipeline applies the equivariant update $X^{(\ell+\frac{1}{2})} = \text{EqvLayer}_\ell(X^{(\ell)})$ and then passes the result through the topology branch, fusing topological features to produce $X^{(\ell+1)}$. Since the clique complex $K(G)$ depends only on the fixed edge set, it is constructed once and reused at every layer. We build it up to a chosen maximum dimension D to limit computational cost (see Appendix G). For a visualization of the architecture, visit Appendix D.1.

Learned filtration. For each simplex $\sigma \in K(G)$, set $f(\sigma) = \rho\Big(\frac{1}{|P(\sigma)|}\sum_{(i,j)\in P(\sigma)} X_{:,i,j}\Big)$, where $\rho : \mathbb{R}^d \to \mathbb{R}$ is a shared two-layer MLP with Xavier initialization. Summing over the full product $\sigma \times \sigma$ is symmetric in the vertices of σ, so combined with the S_n-equivariance of X this yields $f(g \cdot \sigma) = f(\sigma)$ by Lemma 2. We average rather than sum so that filtration values remain comparable across simplex dimensions. Because ρ is shared and applied to each simplex independently, every triangle, tetrahedron, and higher clique gets its own filtration value. The simpler regime in which higher cliques inherit the maximum among their faces (roughly that of [36]) is recovered as a special case but loses information attached to higher simplices, since each clique enters the filtration the moment its last face does.

4.1 Differentiable persistent homology

We use GUDHI [47] for combinatorial persistent homology computation while maintaining gradient flow through the filtration MLP. A valid filtration requires $f(\tau) \leq f(\sigma)$ whenever τ is a face of σ. Since the learned ρ does not enforce this, we apply a differentiable correction processed in order of increasing dimension: for each σ with $\dim(\sigma) \geq 1$, we set

$$\tilde{f}(\sigma) = \max\Big(f(\sigma),\ \max_{\tau \subsetneq \sigma} \tilde{f}(\tau)\Big). \tag{1}$$

We build a GUDHI simplex tree using the values of $\tilde{f}$ as fixed constants (no gradient tracking) to compute the persistence pairs $\{(\sigma_b, \sigma_d)\}$, then compute differentiable lifetimes $\ell_i = \max(\tilde{f}(\sigma_d) - \tilde{f}(\sigma_b), 0)$ retaining gradient tracking through $\tilde{f}$. This standard strategy in differentiable topology layers [10, 12] is valid whenever the persistence pairing is locally constant, which holds for all but a

measure-zero set of filtration values. Essential features (σ_b paired with $\emptyset$) are discarded since their infinite lifetime provides no finite scalar for the lifetime-based vectorization below. Equivariance is unaffected, since tie-merging (Appendix D.2) would handle the birth-representative choice if needed.

4.2 Learned vectorization via attention pooling

Given the lifetimes $\{\ell_i\}_{i=1}^{N_p}$ for a homological dimension p, each lifetime is embedded into $e_i \in \mathbb{R}^{d_v}$ and assigned an attention logit $\alpha_i \in \mathbb{R}$ via two MLPs (Linear–ReLU–Linear), each with hidden dimension d_v. Each homological dimension has a separate pair of MLPs, producing $e_i = \mathrm{MLP}^{(p)}_{\mathrm{embed}}(\ell_i)$ and $\alpha_i = \mathrm{MLP}^{(p)}_{\mathrm{attn}}(\ell_i)$. The graph-level topological vector for dimension p is the weighted sum

$$T^{(p)}_{\mathrm{graph}} = \sum_{i=1}^{N_p} \frac{\exp(\alpha_i)}{\sum_{j=1}^{N_p} \exp(\alpha_j)}\, e_i,$$

which is permutation-invariant since it is a symmetric function of the persistence pairs (when $N_p = 0$ we set $T^{(p)}_{\mathrm{graph}} = \mathbf{0}$). The full topological vector is the concatenation $T_{\mathrm{graph}} = [T^{(0)}_{\mathrm{graph}} \| T^{(1)}_{\mathrm{graph}} \| \cdots] \in \mathbb{R}^{(D+1)\cdot d_v}$.

4.3 Node-level topological features

Broadcasting $T^{(\ell)}$ identically to all pairs loses information on where topological features are located. For the i-th persistence pair (σ_b, σ_d) we therefore define its *involved nodes* as $\mathcal{N}_i = \sigma_b \cup \sigma_d$ (with a tie-merging correction described in Appendix D.2 that ensures equivariance when distinct simplices share a filtration value). We write $\mathcal{N}_i^G$ when the dependence upon a graph G needs to be explicit. A node-level feature vector for each vertex $u \in V$ is then computed by masked attention pooling over the persistence pairs that involve u:

$$t^{(p)}_u = \sum_{i : u \in \mathcal{N}_i} \frac{\exp(\alpha_i)}{\sum_{j : u \in \mathcal{N}_j} \exp(\alpha_j)}\, e_i,$$

using the same embeddings and logits as the graph-level pooling (with $t^{(p)}_u = 0$ when no pair in dimension p involves u).

4.4 Feature fusion

The fusion step incorporates node-level features and uses a gated residual connection. The full per-pair input is $T_{uv} = [T_{\mathrm{graph}} \| t_u \| t_v] \in \mathbb{R}^{3d_T}$, with T_{graph}, t_u, and t_v each passed through LayerNorm before concatenation (LayerNorm normalizes across the feature dimension at each position independently and so commutes with vertex permutations).

Recall from Section 4.4 that the per-pair input is $T_{uv} = [T_{\mathrm{graph}} \| t_u \| t_v]$. Note that $T_{uv} \neq T_{vu}$ in general, as t_u and t_v occupy different positions in the concatenation. This does not cause problems, because the 2nd-order equivariant basis of [36] is defined for general (not necessarily symmetric) tensors, so the later layers handle the non-symmetric inputs correctly.

The update rule is

$$X^{(\ell+1)}_{uv} = X^{(\ell+\frac{1}{2})}_{uv} + \sigma\big(\gamma_\ell(X^{(\ell+\frac{1}{2})}_{uv} \| T_{uv})\big) \odot \psi_\ell\big(X^{(\ell+\frac{1}{2})}_{uv} \| T_{uv}\big),$$

where γ_ℓ and ψ_ℓ are pointwise MLPs, σ is the sigmoid, and $\odot$ is the Hadamard product. The gate γ_ℓ is initialized with bias 2.0 (so $\sigma(2) \approx 0.88$) and the last layer of ψ_ℓ with near-zero weights, so the residual starts near identity while still allowing gradient flow.

Lemma 4. *The gated residual fusion preserves permutation equivariance.*

The proof, which depends on the tie-merging mechanism, is deferred to Appendix D.3. The gated residual formulation does not reduce expressive power relative to either the equivariant network of [36] or the idealized fusion of Section 3. The hypothesis class of the gated model contains the pure equivariant model as a special case, since setting the gate bias to $-\infty$ yields $\sigma(y) \to 0$ and recovers $X^{(\ell+1)} = X^{(\ell+\frac{1}{2})}$ with no topological contribution, preserving the lower bound of Theorem 3.

On the other hand, setting the bias of γ_ℓ to $+\infty$ drives the gate to one, and the update becomes $X_{uv}^{(\ell+1)} = X_{uv}^{(\ell+\frac{1}{2})} + \psi_\ell(X_{uv}^{(\ell+\frac{1}{2})} \| T_{uv})$. Since ψ_ℓ is an MLP, by the universal approximation theorem it can approximate any continuous function, and in particular it can approximate $\psi(X_{uv}\|T^{(\ell)}) - X_{uv}^{(\ell+\frac{1}{2})}$, where ψ is the fusion MLP of Section 3. Substituting yields $X_{uv}^{(\ell+1)} \approx \psi(X_{uv}^{(\ell+\frac{1}{2})} \| T^{(\ell)})$, recovering the idealized fusion.

End-to-end equivariance. Composing the component-wise equivariance and invariance properties yields the following.

Proposition 5 (End-to-end invariance)**.** *Let Φ denote the full model mapping an input tensor $X^{(0)} \in \mathbb{R}^{d\times n\times n}$ to an output $y \in \mathbb{R}^c$, where $X^{(0)}$ is constructed from the graph G as described above. For any permutation $g \in S_n$, $\Phi(g \cdot X^{(0)}) = \Phi(X^{(0)})$.*

The proof, by induction on layers, is deferred to Appendix C.1.

Strict separation beyond 3-WL. We now verify that the strict gain over 3-WL survives every implementation deviation identified above.

Corollary 6. *The implemented model Φ is strictly more expressive than the equivariant network of [36], and hence strictly more expressive than 3-WL: Φ matches the discriminating power of [36] on every graph pair, and there exist non-isomorphic graphs G, G' that are indistinguishable by 3-WL (and hence by [36]) but are distinguished by Φ.*

The proof exhibits explicit parameters that separate the Rook–Shrikhande pair; we defer it to Appendix C.2.

5 Experiments

We augment the 2nd-order invariant graph network of [36] with the topologically-augmented architecture of Section 4. Each equivariant block is followed by a learned filtration MLP ρ_ℓ that produces per-simplex filtration values from the current pairwise tensor. Persistence diagrams are computed via [47], vectorized with the learned attention pooling of Section 4.2, and fused into the pairwise tensor through the gated residual mechanism of Section 4.4. The filtration MLP is Xavier-initialized and trained jointly with the rest of the network. We compute persistence up to homological dimension 1 in clique complexes built up to simplex dimension 2. Two vectorization variants differ only in what each persistence pair contributes to the embedding MLP: **v1** uses the lifetime (death − birth); **v2** uses both birth and persistence as separate coordinates (see Appendix F.1). The base architecture uses three equivariant blocks (d=2 MLP depth per block), invariant max pooling (diagonal and off-diagonal), and fully connected layers ($512 \to 256 \to$ #classes). All topology-augmented models share an identical base architecture and training schedule with the baseline, with only the topology branch parameters added.

Table 1: Test accuracy (%) averaged over 7-fold cross-validation (± population std). Best per column in **bold**; "–" = not yet run.

	NCI1	NCI109	MUTAG	PTC	IMDB-B	PROTEINS
Graphs	4110	4127	188	344	1000	1113
Node labels	37	38	7	22	0	3
Avg. $\lvert V\rvert$	29.8	29.6	17.9	25.5	19.7	39.1
Baseline	77.45±1.38	75.99±1.21	86.16±6.19	62.78±5.76	70.4±3.31	**76.73±2.02**
Ours (v1)	78.98±1.00	78.51±1.01	87.26±6.79	64.84±5.64	–	–
Ours (v2)	**80.25±1.05**	**80.19±1.15**	**88.84±7.12**	**66.6±5.92**	70.5±3.92	76.37±2.75

We evaluate on six graph-classification benchmarks from the TU Dortmund repository [53]: four molecular datasets with categorical node labels — NCI1 and NCI109 (anti-cancer activity), MUTAG (mutagenicity), PTC (rodent carcinogenicity); one protein-structure dataset, PROTEINS (3 node labels; enzymes vs. non-enzymes); and one social-network dataset without node features, IMDB-B (movie-genre classification from actor-collaboration graphs). We run 7-fold stratified cross-validation, select the epoch with best mean validation accuracy, and report mean test accuracy and population standard deviation across folds. Hyperparameter selection covers learning rate $\in \{10^{-4}, 5\times10^{-4}, 10^{-3}\}$, decay factor $\in [0.5, 0.95]$ applied every 20 epochs, and topology branch widths `topo_num_stats` $\in \{8, 16, 32\}$, `topo_hidden_dim` $\in \{8, 16, 32\}$. Experiments were conducted on NVIDIA P100, A100, and RTX 4000 Ada GPUs with fixed seeds. We focus comparisons of v1 against v2 on the four molecular datasets, where topology produces a meaningful signal; on IMDB-B and PROTEINS, the diagnostic analysis (Appendix E) shows that the filtration branch produces uninformative signals regardless of vectorization choice, so we report only v2 there.

5.1 Results

Results are collected in Table 1. Persistent homology consistently improves over the baseline on all four molecular benchmarks, with gains largest on NCI109 (+4.2%) and PTC (+3.8%) and moderate on NCI1 (+2.8%) and MUTAG (+2.7%). The richer birth–persistence representation (v2) outperforms the lifetime-only representation (v1) on every molecular dataset, suggesting that absolute filtration scale carries discriminative information beyond what lifetime alone encodes; this is consistent with the chemical intuition that ring formation occurs at characteristic scales. On IMDB-B (no node features) topology provides only a +0.1% gain, and on PROTEINS the baseline edges out the topology model by 0.4% (within the standard deviation of both). The diagnostic analysis in Appendix E traces both outcomes to filtration informativeness: on IMDB-B the layer-0 filtration is degenerate (zero finite persistence pairs throughout training), and on PROTEINS the filtration MLP enters a pathological regime in which the empirical spread blows up by nearly 30× during training, so the topology branch overfits in the absence of an informative filtration signal.

5.2 Empirical Validation of Corollary 6

Corollary 6 establishes the existence of parameters under which our model separates the Rook–Shrikhande pair, but the construction is specific and hand-tuned. A natural question is whether this separation persists for a randomly initialized model, and whether it extends beyond a single witness pair. To address both, we directly evaluate our model on five pairs of non-isomorphic, cospectral, strongly regular graphs drawn from three classical families: Rook(4,4) and the Shrikhande graph [46]

Table 2: Output differences on five pairs of cospectral, 3-WL equivalent strongly regular graphs at random initialization. Baseline differences are at floating-point noise; topology-augmented differences exceed the threshold of 10^{-5} on every pair.

Pair	Baseline diff.	Topology diff.	Separated
Rook(4,4) vs. Shrikhande	1.5×10^{-8}	1.8×10^{-1}	✓
$T(8)$ vs. Chang$_1$	0	1.1×10^{-1}	✓
$T(8)$ vs. Chang$_2$	0	1.2×10^{-1}	✓
$T(8)$ vs. Chang$_3$	3.0×10^{-8}	6.5×10^{-2}	✓
Paley(25) vs. $L_3(5)$	3.0×10^{-8}	1.1×10^{-3}	✓
Paley(49) vs. Peisert(49)	6.0×10^{-8}	6.4×10^{-3}	✓

at parameters $\mathrm{srg}(16, 6, 2, 2)$; the triangular graph $T(8) = L(K_8)$ and the three Chang graphs [15, 16] at $\mathrm{srg}(28, 12, 6, 4)$; and Paley(25) versus the Latin-square graph $L_3(5)$ at $\mathrm{srg}(25, 12, 5, 6)$ along with Paley(49) versus Peisert(49) [42, 43] at $\mathrm{srg}(49, 24, 11, 12)$. All five pairs are 3-WL equivalent [11], so any architecture whose expressive power is bounded by 3-WL must produce identical outputs on each pair.

Setup. For each pair (G_1, G_2), we run a forward pass on both graphs through the baseline 2-IGN of [36] and through our topology-augmented model, using the same random initialization across the two graphs in each pair and matched architectures otherwise (block features $[64, 64]$, MLP depth 2; the topology branch adds max_ph_dim $= 2$, max_simplex_dim $= 3$, num_stats $= 16$, hidden_dim $= 32$). We declare a pair *separated* by a model if the maximum coordinate-wise difference of the two outputs exceeds 10^{-5}, comfortably above the floating-point noise we observe on the baseline.

Results. Table 2 summarizes the outcome. The baseline produces outputs that agree to within numerical precision on every pair, consistent with its 3-WL bound. Our topology-augmented model separates every pair, with output differences ranging from 10^{-3} to 10^{-1}, four to seven orders of magnitude above the baseline. Three observations are worth highlighting. First, the separation does not require training: a randomly initialized topology branch already produces distinct persistence-derived features, since the structural differences are encoded in the simplicial complex itself rather than in any learned parameter. Second, the magnitude of the topology-driven separation correlates with the degree to which the underlying clique complexes differ; the Rook–Shrikhande pair has the largest gap (an 8-fold difference in 3-simplex counts), and the Chang pairs follow with substantial 3-simplex differences. Third, the Paley(25) versus $L_3(5)$ pair is a stringent test: both graphs have identical simplex counts in every dimension and identical finite persistence pair counts under the degree filtration; the fact that our model still separates them (with the smallest margin of the five) indicates that the discrimination arises from the interaction between the equivariant pairwise tensor and the learned filtration MLP rather than from raw persistence statistics alone. Together with Corollary 6, these results convert the theoretical expressivity gain into a directly verifiable empirical property: across multiple SR families spanning orders 16 to 49, our architecture distinguishes graph pairs that no 3-WL bounded model can.

5.3 Diagnostic Analysis (Summary)

The TU benchmark results show a non-uniform pattern: topology yields consistent gains on the four molecular datasets, is roughly neutral on IMDB-B, and slightly worsens performance on

PROTEINS. We instrument the trained models with three families of diagnostics—per-layer gate activation $\sigma(\gamma_\ell)$, persistence-pair counts and filtration-value statistics, and the fraction of pairs modified by tie-merging—and find that the deep-layer gate at convergence acts as a direct readout of filtration informativeness. It separates the five datasets into three regimes: *stable commitment* (high mean, low std; MUTAG, PTC, NCI1, where accuracy improves), *noisy commitment* (high mean, moderate std; IMDB-B, where accuracy is essentially unchanged), and *indecision* (low mean, high std; PROTEINS, where accuracy regresses). Filtration quality, gate behavior, and accuracy track one another on a per-dataset basis. The full breakdown—gate trajectories, regime scatter, filtration spread, persistence-pair counts, training curves, and tie-merging statistics—is deferred to Appendix E.

6 Conclusion

We introduced a class of topologically-augmented equivariant graph neural networks that combine k-order tensor representations with persistent homology via equivariant filtrations on clique complexes, and provided a rigorous expressivity analysis showing that the architecture is at least as expressive as k-WL and strictly more expressive than 3-WL. Beyond the idealized model, we verified that the practical design choices in our implementation—learned attention-pooling, gated residual fusion, tie-merged node-level features, and the birth–persistence variant of Appendix F—preserve both end-to-end permutation invariance and the strict separation of the Rook–Shrikhande witness pair, closing the theory–practice gap highlighted by [36]. Our experiments validate both expressivity claims directly: at random initialization the model separates five pairs of cospectral, 3-WL-equivalent strongly regular graphs spanning orders 16 to 49, with output differences four to seven orders of magnitude above baseline floating-point noise, and on the four molecular TU Dortmund benchmarks it yields consistent +2.7% to +3.8% accuracy gains over the equivariant baseline of [36]. These results establish combinatorial and topological methods as genuinely complementary within a single learnable framework, and provide a foundation for future work on integrating multi-scale topological structure into expressive graph learning.

References

[1] R. Ballester, C. Casacuberta, and S. Escalera. Topological data analysis for neural network analysis: A comprehensive survey. *arXiv preprint arXiv:2312.05840*, 2023.

[2] R. Ballester and B. Rieck. On the expressivity of persistent homology in graph learning. *arXiv preprint arXiv:2302.09826*, 2023.

[3] F. Barbero, C. Bodnar, H. Sáez de Ocariz Borde, M. M. Bronstein, P. Veličković, and P. Liò. Sheaf neural networks with connection laplacians. *arXiv preprint arXiv:2206.08702*, 2022.

[4] U. Bauer, T. B. Masood, B. Giunti, G. Houry, M. Kerber, and A. Rathod. Keeping it sparse: Computing persistent homology revisited, 2024.

[5] C. Bodnar, F. Di Giovanni, B. P. Chamberlain, P. Liò, and M. M. Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns. 2022.

[6] C. Bodnar, F. Frasca, Y. G. Wang, N. Otter, G. Montúfar, P. Liò, and M. Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks, 2021.

[7] G. Bouritsas, F. Frasca, S. Zafeiriou, and M. M. Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):657–668, 2022.

[8] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[9] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[10] R. Brüel-Gabrielsson, B. J. Nelson, A. Dwaraknath, P. Skraba, L. J. Guibas, and G. Carlsson. A topology layer for machine learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1553–1563. PMLR, 2020.

[11] J.-Y. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.

[12] M. Carrière, F. Chazal, M. Glisse, Y. Ike, H. Kannan, and Y. Umeda. Optimizing persistent homology based functions. In *International Conference on Machine Learning*, pages 1294–1303. PMLR, 2021.

[13] M. Carrière, F. Chazal, Y. Ike, T. Lacombe, M. Royer, and Y. Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In *International Conference on Artificial Intelligence and Statistics*, pages 2786–2796. PMLR, 2020.

[14] G. Cesa and A. Behboodi. Algebraic topological networks via the persistent local homology sheaf. In *Symmetry and Geometry in Neural Representations (Extended Abstract Track, NeurIPS 2023)*. Qualcomm AI Research, 2023.

[15] L.-C. Chang. The uniqueness and nonuniqueness of the triangular association scheme. *Science Record*, 3:604–613, 1959.

[16] L.-C. Chang. Association schemes of partially balanced block designs with parameters v= 28, n1= 12, n2= 15 and p2 11= 4, sci. *Record*, 4:12–18, 1960.

[17] Y. Chen, B. Coskunuzer, and Y. Gel. Topological relational learning on graphs. *Advances in neural information processing systems*, 34:27029–27042, 2021.

[18] Y. Chen and Y. R. Gel. Topological pooling on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7096–7103, 2023.

[19] Z. Chen, L. Chen, S. Villar, and J. Bruna. Can graph neural networks count substructures? *Advances in neural information processing systems*, 33:10383–10395, 2020.

[20] T. K. Dey and Y. Wang. *Computational topology for data analysis*. Cambridge University Press, 2022.

[21] S. Ebli, M. Defferrard, and G. Spreemann. Simplicial neural networks. In *Topological Data Analysis and Beyond Workshop at Neural Information Processing Systems (NeurIPS) 2020*, 2020.

[22] H. Edelsbrunner and J. Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[23] M. Grohe. The logic of graph neural networks. *CoRR*, abs/2104.14624, 2021.

[24] J. Hansen and T. Gebhart. Sheaf neural networks. *arXiv preprint arXiv:2012.06333*, 2020.

[25] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

[26] F. Hensel, M. Moor, and B. Rieck. A survey of topological machine learning methods. *Frontiers in Artificial Intelligence*, 4:681108, 2021.

[27] C. Hofer, F. Graf, B. Rieck, M. Niethammer, and R. Kwitt. Graph filtration learning. In *International Conference on Machine Learning*, pages 4314–4323. PMLR, 2020.

[28] C. Hofer, R. Kwitt, M. Niethammer, and A. Uhl. Deep learning with topological signatures. *Advances in neural information processing systems*, 30, 2017.

[29] C. D. Hofer, R. Kwitt, and M. Niethammer. Learning representations of persistence barcodes. *Journal of Machine Learning Research*, 20(126):1–45, 2019.

[30] M. Horn, E. D. Brouwer, M. Moor, Y. Moreau, B. Rieck, and K. Borgwardt. Topological graph neural networks. In *International Conference on Learning Representations*, 2022.

[31] J. Immonen, A. Souza, and V. Garg. Going beyond persistent homology using persistent homology. *Advances in neural information processing systems*, 36:63150–63173, 2023.

[32] J. Immonen, A. H. Souza, and V. Garg. Going beyond persistent homology using persistent homology, 2023.

[33] C. K. Joshi, C. Bodnar, S. V. Mathis, T. Cohen, and P. Lio. On the expressive power of geometric graph neural networks. In *International conference on machine learning*, pages 15330–15355. PMLR, 2023.

[34] N. Keriven and G. Peyré. Universal invariant and equivariant graph neural networks, 2019.

[35] P. Li and J. Leskovec. The expressive power of graph neural networks. In *Graph Neural Networks: Foundations, Frontiers, and Applications*, pages 63–98. Springer, 2022.

[36] H. Maron, H. Ben-Hamu, H. Serviansky, and Y. Lipman. Provably powerful graph networks. *Advances in neural information processing systems*, 32, 2019.

[37] H. Maron, H. Ben-Hamu, N. Shamir, and Y. Lipman. Invariant and equivariant graph networks. *arXiv preprint arXiv:1812.09902*, 2018.

[38] N. Milosavljević, D. Morozov, and P. Skraba. Zigzag persistent homology in matrix multiplication time. In *Proceedings of the twenty-seventh Annual Symposium on Computational Geometry*, pages 216–225, 2011.

[39] E. Minamitani, I. Obayashi, K. Shimizu, and S. Watanabe. Persistent homology-based descriptor for machine-learning potential of amorphous structures. *The Journal of Chemical Physics*, 159(8), 2023.

[40] C. Morris, Y. Lipman, H. Maron, B. Rieck, N. M. Kriege, M. Grohe, M. Fey, and K. Borgwardt. Weisfeiler and leman go machine learning: The story so far. *ACM SIGACT News*, 53(2):71–99, 2022. Foundations of the Weisfeiler–Leman hierarchy and its relation to GNN expressivity.

[41] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

[42] R. E. Paley. On orthogonal matrices. *Journal of mathematics and Physics*, 12(1-4):311–320, 1933.

[43] W. Peisert. All self-complementary symmetric graphs. *Journal of Algebra*, 240(1):209–229, 2001.

[44] B. Rieck, C. Bock, and K. Borgwardt. A persistent weisfeiler-lehman procedure for graph classification. In *International Conference on Machine Learning*, pages 5448–5458. PMLR, 2019.

[45] R. Sato. A survey on the expressive power of graph neural networks. *arXiv preprint arXiv:2003.04078*, 2020.

[46] S. S. Shrikhande. The uniqueness of the L_2 association scheme. *The Annals of Mathematical Statistics*, 30(3):781–798, 1959.

[47] The GUDHI Project. GUDHI: Geometry understanding in higher dimensions, 2014.

[48] R. Turkeš, G. Montúfar, and N. Otter. On the effectiveness of persistent homology, 2023.

[49] Y. Verma, A. H. Souza, and V. Garg. Topological neural networks go persistent, equivariant, and continuous. *arXiv preprint arXiv:2406.03164*, 2024.

[50] Q. Wang, D. Z. Chen, A. Wijesinghe, S. Li, and M. Farhan. $\mathscr{N}$-WL: A new hierarchy of expressivity for graph neural networks. In *The Eleventh International Conference on Learning Representations*, 2023.

[51] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[52] Z. Yan, T. Ma, L. Gao, Z. Tang, Y. Wang, and C. Chen. Neural approximation of graph topological features. *Advances in neural information processing systems*, 35:33357–33370, 2022.

[53] P. Yanardag and S. V. Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1365–1374, 2015.

[54] X. Ye, F. Sun, and S. Xiang. Treph: A plug-in topological layer for graph neural networks. *Entropy*, 25(2):331, 2023.

[55] C. Ying, X. Zhao, and T. Yu. Boosting graph pooling with persistent homology. *Advances in Neural Information Processing Systems*, 37:19087–19113, 2024.

[56] B. Zhang, G. Feng, Y. Du, D. He, and L. Wang. A complete expressiveness hierarchy for subgraph gnns via subgraph weisfeiler-lehman tests. In *International Conference on Machine Learning*, pages 41019–41077. PMLR, 2023.

[57] S. Zhang, S. Mukherjee, and T. K. Dey. Gefl: Extended filtration learning for graph classification. In *Learning on Graphs Conference*, pages 16–1. PMLR, 2022.

[58] Q. Zhao and Y. Wang. Learning metrics for persistence-based summaries and applications for graph classification. *Advances in neural information processing systems*, 32, 2019.

[59] Q. Zhao, Z. Ye, C. Chen, and Y. Wang. Persistence enhanced graph neural network. In *International Conference on Artificial Intelligence and Statistics*, pages 2896–2906. PMLR, 2020.

A Related Works

Topological data analysis for graph learning. Topological data analysis (TDA) has been applied in a variety of machine learning settings; see [26] for a general overview and [1] for applications in neural networks. In graph learning, persistent homology has been used to construct invariant descriptors, pooling operators, and regularization schemes [13, 17, 28, 27, 29, 30, 44, 52, 54, 58, 59, 20, 22, 55, 32, 39]. The PersLay framework of [13] introduced a general neural network layer for persistence diagrams, establishing that many existing vectorization schemes (persistence images, landscapes, Betti curves) can be expressed as special cases of a single learnable architecture. More recently, persistent homology has been used as pooling layers [18, 55], readout layers [57], and regular layers within GNN architectures [30, 49, 54].

Expressivity of GNNs and the WL hierarchy. The foundational connection between GNN expressivity and the Weisfeiler–Lehman hierarchy was established by [51] and [41], who showed that standard message-passing GNNs are at most as expressive as 1-WL. Extensions to higher-order architectures have demonstrated that equivariant tensor-based networks can match the expressive power of k-WL [36, 37]. In particular, [36] proved that k-order invariant graph networks are as powerful as k-WL for $k \geq 2$, and that a practical 2-order model with matrix multiplication achieves 3-WL expressiveness. This latter result is especially relevant to our work, as our equivariant branch employs a similar parameterization (parallel MLPs with matrix multiplication) and we recover their 2-WL simulation as a special case. See [40] for a comprehensive survey of the WL hierarchy and its relation to GNN expressivity, and [45, 23, 35] for a broader survey of expressive power in GNNs. More recently, [50] introduce the Neighbourhood WL ($\mathcal{N}$-WL) hierarchy, arguing that classical k-WL is not well-suited for characterizing GNN expressivity and proposing a subgraph-based alternative aligned with practical architectures. Building on this perspective, [56] develop a complete expressiveness hierarchy for subgraph GNNs via Subgraph WL tests, showing that such models fall into distinct equivalence classes with strictly increasing expressive power.

Expressivity of persistent homology. The expressivity of topological methods for graph learning has been analyzed in several recent works. [30] proved that persistent homology is at least as expressive as 1-WL when combined with an appropriate filtration, and [2] extended these results to the full WL hierarchy. Their Theorem 1 shows that for $k \geq 2$, there exists an equivariant filtration generator whose zero-dimensional persistence diagrams are at least as expressive as k-FWL, albeit using a type-2 filtration that considers all possible cliques. They further showed that the Vietoris–Rips filtration is *incomparable* to 1-WL: it can detect connected components that 1-WL misses, while 1-WL can distinguish certain path and cycle graphs that Vietoris–Rips cannot. This incomparability is a key motivation for our hybrid approach. [31] also analyzed the expressivity of topological methods, and [48] investigated the effectiveness of persistent homology from a complementary angle.

Simplicial and higher-order neural networks. The extension of graph neural networks to simplicial complexes was pioneered by [21], who defined simplicial convolutions via the Hodge Laplacian and demonstrated their utility for missing data imputation on coauthorship complexes. Their framework processes signals on p-simplices (not just vertices) using spectral methods rooted in

Hodge–de Rham theory, providing a principled notion of locality and shift-invariance on topological spaces. While their work focuses on signal processing on fixed simplicial complexes rather than on graph isomorphism expressivity, the Hodge-theoretic perspective—particularly the observation that the kernel of the k-Laplacian is isomorphic to k-cohomology—is closely related to the topological features our model captures via persistent homology. Other related work includes sheaf neural networks [24, 3, 5], which equip graphs with algebraic structures (sheaves) to model heterogeneous data, and algebraic topological networks [14], which use persistent local homology sheaves.

Substructure counting and geometric expressivity. Complementary approaches to enhancing GNN expressivity include substructure counting methods [19, 7], which augment node features with counts of specific subgraph patterns, and geometric GNNs [33], which leverage positional and structural encodings. While substructure counting can provably enhance expressivity for specific motifs, it requires pre-specifying which patterns to count and does not naturally capture multi-scale topological features. Our approach is more general in this regard: persistent homology detects topological features across all scales simultaneously, without requiring explicit enumeration of substructure types.

Positioning of our work. Our work connects the equivariant network line of research [36, 37] with the topological expressivity line [2, 30] by integrating persistent homology into higher-order equivariant graph networks. In contrast to prior work that studies either WL-based expressivity or topological descriptors in isolation, we provide a joint analysis that captures both local combinatorial refinement and global topological structure within a single architecture. Unlike [21], who process fixed cochains on simplicial complexes, our model learns filtrations from equivariant tensor representations and feeds the resulting topological features back into the equivariant branch, enabling end-to-end training. Unlike [2], who analyze persistent homology as a standalone descriptor, we study its interaction with neural architectures and prove that this integration preserves and extends the expressivity guarantees of both components.

B Deferred Details from Section 3

B.1 Filtration Function Figure

We provide a visualization of our filtration function pipeline applied to a small graph.

B.2 Proof of Lemma 2 (Equivariance of the Filtration)

Proof. Let $g \in S_n$ act on simplices by relabelling vertices: $g \cdot \sigma = \{g(v) : v \in \sigma\}$. S_n acts on multi-indices entrywise, $g(\boldsymbol{i}) = (g(i_1), \dots, g(i_k))$, and on tensors by $(g \cdot X)_{\boldsymbol{i},j} = X_{g^{-1}(\boldsymbol{i}),j}$. Specializing to $k = 2$ and writing $\boldsymbol{i} = (i_1, i_2)$ with the feature index suppressed,

$$(g \cdot X)_{i_1,i_2} \;=\; X_{g^{-1}(i_1),g^{-1}(i_2)} \qquad \text{for all } i_1, i_2 \in [n].$$

Reindexing $i_1 \mapsto g(i_1)$ and $i_2 \mapsto g(i_2)$ on both sides yields the equivalent identity

$$(g \cdot X)_{g(i_1),g(i_2)} \;=\; X_{i_1,i_2} \qquad \text{for all } i_1, i_2 \in [n], \tag{2}$$

which holds unconditionally for every $g \in S_n$. It is purely the definition of the action, written index-side-out.

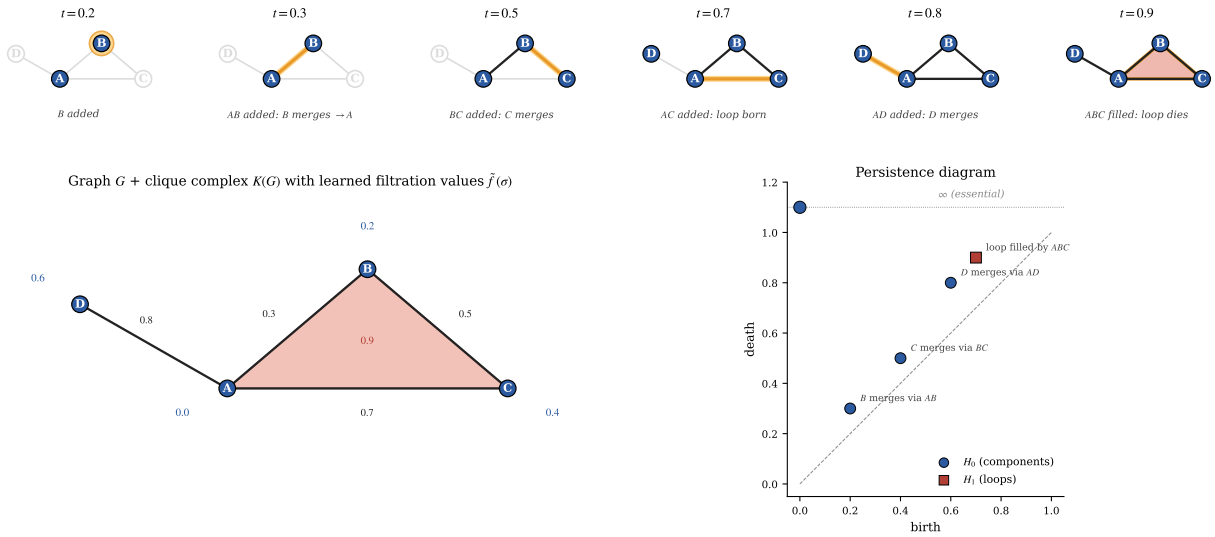


Figure 1: The learned filtration on a small example. Graph G (triangle ABC plus pendant edge AD); the bottom-left panel shows its clique complex $K(G)$ with the learned, monotonicity-corrected filtration value $\tilde{f}(\sigma)$ labelled on every simplex. The top row visualises the sublevel-set filtration $K_t = \{\sigma : \tilde{f}(\sigma) \leq t\}$ at six thresholds; the simplex newly added at each t is highlighted in gold. Components are born when an isolated vertex appears and die when an edge merges them ($t = 0.3, 0.5, 0.8$); the 1-cycle is born when edge AC closes the triangle's boundary ($t = 0.7$) and dies when the 2-simplex ABC fills it ($t = 0.9$). The bottom-right panel shows the resulting persistence diagram, with H_0 component-merge pairs (circles), the essential H_0 class at ∞, and the single H_1 pair at $(0.7, 0.9)$ (square).

The set of ordered pairs in the relabelled simplex is the image of $P(\sigma)$ under g:

$$P(g \cdot \sigma) \;=\; \{(g(i),\, g(j)) : (i,j) \in P(\sigma)\}.$$

Reindexing the sum over $P(g \cdot \sigma)$ by this bijection and applying (2),

$$\sum_{(i',j') \in P(g\cdot\sigma)} (g \cdot X)_{i',j'} \;=\; \sum_{(i,j)\in P(\sigma)} (g \cdot X)_{g(i),\, g(j)} \;=\; \sum_{(i,j)\in P(\sigma)} X_{i,j},$$

where the equality holds componentwise in the d-dimensional feature index.

Since ρ_ℓ is the same MLP regardless of which tensor or simplex it is applied to,

$$f_\ell(g \cdot X,\, g \cdot \sigma) \;=\; \rho_\ell\left(\frac{1}{|P(g\cdot\sigma)|} \sum_{(i',j')\in P(g\cdot\sigma)} (g \cdot X)_{i',j'}\right) \;=\; \rho_\ell\left(\frac{1}{|P(\sigma)|} \sum_{(i,j)\in P(\sigma)} X_{i,j}\right) \;=\; f_\ell(X,\, \sigma),$$

Thus the raw learned score f_ℓ is equivariant.

It remains to show that the corrected filtration $\tilde{f}_\ell$ is equivariant. We prove this by induction on $\dim(\sigma)$. If $\dim(\sigma) = 0$, then $\tilde{f}_\ell(X, \sigma) = f_\ell(X, \sigma)$, so the claim follows from the equivariance of the raw score.

Assume the claim holds for all simplices of dimension strictly less than p, and let $\dim(\sigma) = p$. Since g is a bijection on vertices, the map $\tau \mapsto g \cdot \tau$ is a bijection from the proper faces of σ to the proper faces of $g \cdot \sigma$. Hence

$$\max_{\tau' \subsetneq g\cdot\sigma} \tilde{f}_\ell(g \cdot X, \tau') = \max_{\tau \subsetneq \sigma} \tilde{f}_\ell(g \cdot X, g \cdot \tau).$$

By the induction hypothesis, for every $\tau \subsetneq \sigma$,

$$\tilde{f}_\ell(g \cdot X, g \cdot \tau) = \tilde{f}_\ell(X, \tau),$$

since $\dim(\tau) < \dim(\sigma)$. Therefore,

$$\max_{\tau' \subsetneq g\cdot\sigma} \tilde{f}_\ell(g \cdot X, \tau') = \max_{\tau \subsetneq \sigma} \tilde{f}_\ell(X, \tau).$$

Combining this with $f_\ell(g \cdot X, g \cdot \sigma) = f_\ell(X, \sigma)$ and the recursive definition of the max-correction,

$$\tilde{f}_\ell(g{\cdot}X, g{\cdot}\sigma) = \max\left(f_\ell(g \cdot X, g \cdot \sigma), \max_{\tau' \subsetneq g\cdot\sigma} \tilde{f}_\ell(g \cdot X, \tau')\right) = \max\left(f_\ell(X, \sigma), \max_{\tau \subsetneq \sigma} \tilde{f}_\ell(X, \tau)\right) = \tilde{f}_\ell(X, \sigma).$$

This proves the equivariance of the corrected filtration. □

B.3 Proof of Theorem 3 (Expressivity guarantee)

Proof. The proof proceeds by showing that the equivariant branch alone can simulate k-WL, and that the topological branch does not reduce expressivity.

Part 1: The equivariant branch simulates k-WL. A k-order equivariant network can encode the multisets $\{\{X_{\mathbf{j}} : \mathbf{j} \in N_\ell(\mathbf{i})\}\}$ for a given tensor $X \in \mathbb{R}^{d\times n^k}$, where $N_\ell(\mathbf{i}) = \{(i_1, \ldots, i_{\ell-1}, v, i_{\ell+1}, \ldots, i_k) : v \in [n]\}$. As shown by [36], these multisets are the only non-trivial component in the k-WL update rule. By the universality of MLPs on finite domains, for each round of k-WL there exist equivariant layer parameters that exactly reproduce the color refinement. Inductively, a stack of L equivariant layers simulates L rounds of k-WL.

For $k = 2$, this recovers the result of [36]: a 2nd-order equivariant network with matrix multiplication can simulate the Folklore WL refinement

$$\chi^{(t+1)}(u,v) = \mathrm{HASH}\Big(\chi^{(t)}(u,v),\ \{\!\{\big(\chi^{(t)}(u,w),\ \chi^{(t)}(w,v)\big) : w \in V\}\!\}\Big),$$

since the matrix product $\sum_{w\in V} \mathrm{MLP}_1(X)_{uw}\,\mathrm{MLP}_2(X)_{wv}$ aggregates over all $w \in V$, mirroring the multiset aggregation in 2-WL.

Part 2: The topological branch does not reduce expressivity. When the topological branch is active, each layer computes

$$X_{uv}^{(\ell+1)} = \psi_\ell\big(X_{uv}^{(\ell+\frac{1}{2})} \,\|\, T^{(\ell)}\big).$$

Since $T^{(\ell)}$ is a fixed (graph-level) vector that is identical for all pairs (u,v), the MLP ψ_ℓ can choose to ignore it entirely—for instance, by setting the weights corresponding to the $T^{(\ell)}$ input channels to zero. In this case, ψ_ℓ reduces to a function of $X_{uv}^{(\ell+\frac{1}{2})}$ alone, recovering the pure equivariant update. Therefore, the topology-augmented hypothesis class *contains* the pure equivariant hypothesis class, and by Part 1 it is at least as expressive as k-WL. □

C Deferred Proofs from Section 4

C.1 Proof of Proposition 5 (End-to-end invariance)

Proof. We argue by induction on layers.

Base case ($\ell = 0$). The input tensor $X^{(0)}$ is equivariant since permuting vertex labels permutes the rows and columns of each channel, which is the standard S_n action on 2-tensors.

Inductive step. Suppose $X^{(\ell)}$ is equivariant. Then $X_{\mathrm{pre}}^{(\ell)} = X^{(\ell)}$ is equivariant by assumption, and $X^{(\ell+\frac{1}{2})} = \mathrm{EqvLayer}_\ell(X^{(\ell)})$ is a 2nd-order equivariant map by [36]. The filtration function $\tilde{f}$ computed from $X_{\mathrm{pre}}^{(\ell)}$ is equivariant by Lemma 2, and the persistence diagram is therefore invariant by Proposition 1. The graph-level vector T_{graph} is invariant as a concatenation of invariant vectors across homological dimensions and the node-level features t_u are equivariant by Proposition 9. The gated residual fusion produces equivariant $X^{(\ell+1)}$ by Lemma 4.

Readout. The final readout sums the diagonal of the last equivariant tensor and passes the result through an MLP. The diagonal sum is a 2nd-order invariant map by [37] and the MLP is a pointwise (permutation-independent) function. Thus, their composition is invariant, yielding $\Phi(g \cdot X^{(0)}) = \Phi(X^{(0)})$. □

C.2 Proof of Corollary 6 (Strict separation beyond 3-WL)

Proof. We exhibit explicit parameters of Φ that (i) match the discriminating power of the equivariant network of [36] under matrix multiplication, namely 3-WL, and (ii) separate the 3-WL-equivalent rook–Shrikhande pair.

For (i), set the bias of every gate γ_ℓ to $-M$ and its weights to zero. Then $\sigma(\gamma_\ell(\cdot)) = \sigma(-M) \to 0$ uniformly as $M \to \infty$, so $X^{(\ell+1)}$ converges uniformly to $X^{(\ell+\frac{1}{2})} = \mathrm{EqvLayer}_\ell(X^{(\ell)})$. By continuity of the readout, Φ matches the pure equivariant network of [36], which under matrix multiplication is 3-WL expressive.

For (ii), let $R = K_4 \square K_4$ and let S be the Shrikhande graph. Both are strongly regular with parameters $(16, 6, 2, 2)$. By [7], 2-FWL fails to distinguish strongly regular graphs with identical parameters, and explicitly cite the Rook–Shrikhande pair as an example, separated by 4-clique counts

but not by 2-FWL. Since 2-FWL is equivalent to 3-WL [36], the pair is 3-WL-indistinguishable. The graph R contains exactly 8 four-cliques (its 4 rows and 4 columns), while S has clique number 3 [6], so $K(R)$ has 8 three-simplices and $K(S)$ has none. Distinct 4-cliques of R meet in at most one vertex, so they share no edges and no triangles; each triangle of R lies in a unique 4-clique, and each edge of R lies in two triangles, both within the same 4-clique.

Construct an explicit parameter setting as follows. Set $L = 1$, $D = 3$, and $d_0 = d_1 = d_v = 1$. Take $X^{(0)}_{0,i,j} = A_{i,j}$ (the adjacency matrix), $\mathrm{EqvLayer}_1 = \mathrm{Id}$ (achievable by activating only the identity basis operator of [37] and zeroing the matmul branch), and $\rho_1 = \mathrm{Id}$ (a two-layer ReLU MLP on $\mathbb{R}_{\geq 0}$). For any p-simplex σ, the off-diagonal entries of $P(\sigma) = \sigma \times \sigma$ are all edges, so $f_1(\sigma) = p(p+1)/(p+1)^2 = p/(p+1)$; vertices, edges, triangles, and tetrahedra thus receive filtration values $0, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}$ respectively, identical in $K(R)$ and $K(S)$ and strictly increasing in dimension, so the non-decreasing correction acts trivially.

We now compute the dimension-2 persistence diagrams of $K(R)$ and $K(S)$, denoted by $D_2(R)$ and $D_2(S)$. In $K(R)$, the 32 triangles enter at $\frac{2}{3}$ and the 8 tetrahedra at $\frac{3}{4}$. The four face triangles of any 4-clique sum to a ∂_2-cycle (each edge appears in two of the four faces), yielding one independent 2-cycle per 4-clique. These eight are the only 2-cycles, since each edge of R lies in exactly two triangles, both within the same 4-clique, forcing any 2-cycle to consist of full 4-clique boundary spheres, so $\dim\ker\partial_2 = 8$. The boundary in C_2 of each tetrahedron is the corresponding 2-sphere, and distinct tetrahedra have disjoint triangle-supports, so ∂_3 surjects onto $\ker\partial_2$ and each tetrahedron kills one H_2 class. Therefore $D_2(R) = \{(\frac{2}{3}, \frac{3}{4})\}^8$, all eight pairs of lifetime $\frac{1}{12}$. In $K(S)$, the clique number is 3, so $C_3(K(S)) = 0$ and $\partial_3 = 0$; every class in $\ker\partial_2(K(S))$ is therefore born but never killed, i.e. essential. Since the architecture discards essential features (Section 4.1), $D_2(S) = \varnothing$.

We now choose the remaining parameters so that the difference between $D_2(R)$ and $D_2(S)$ propagates to a difference between $\Phi(R)$ and $\Phi(S)$. We vectorize with $\mathrm{MLP}^{(2)}_{\mathrm{attn}} \equiv 0$ (uniform softmax weights $\frac{1}{8}$) and $\mathrm{MLP}^{(2)}_{\mathrm{embed}} = \mathrm{Id}$, with $\mathrm{MLP}^{(p)}_{\mathrm{attn}}, \mathrm{MLP}^{(p)}_{\mathrm{embed}} \equiv 0$ for $p \in \{0, 1, 3\}$. We track $T_{\mathrm{graph}}(G)$ dimension by dimension. For $p \in \{0, 1\}$, the choice $\mathrm{MLP}^{(p)}_{\mathrm{embed}} \equiv 0$ forces every per-pair embedding $e_i = 0$, so $T^{(p)}_{\mathrm{graph}}(G) = 0$ on both graphs regardless of the persistence diagram in that dimension. For $p = 3$, $K(R)$ contains no 4-simplices and $K(S)$ contains no 3-simplices, so neither produces finite H_3 persistence pairs, and the empty-set convention of Section 4.2 yields $T^{(3)}_{\mathrm{graph}}(G) = 0$ on both graphs. For $p = 2$, $K(S)$ produces no finite pairs (its H_2 classes are essential and discarded), so $T^{(2)}_{\mathrm{graph}}(S) = 0$ by the same convention; on $K(R)$, the eight H_2 pairs at $(\frac{2}{3}, \frac{3}{4})$ each give $e_i = \frac{1}{12}$ under $\mathrm{MLP}^{(2)}_{\mathrm{embed}} = \mathrm{Id}$, and the uniform softmax weights $\frac{1}{8}$ yield $T^{(2)}_{\mathrm{graph}}(R) = \frac{1}{12}$. Combining, $T_{\mathrm{graph}}(R) = (0, 0, \frac{1}{12}, 0)$ and $T_{\mathrm{graph}}(S) = \mathbf{0} \in \mathbb{R}^4$. Every vertex of R lies in some triangle and some tetrahedron, so the tie-merged involved-node set $V_{2,2/3} \cup V_{3,3/4}$ equals V for every dimension-2 pair, yielding $t^{(2)}_u(R) = \frac{1}{12}$ for every u and $t^{(2)}_u(S) = 0$. After LayerNorm, the third coordinate of $T_{\mathrm{graph}}(R)$ equals $(\frac{1}{12} - \frac{1}{48})/\sqrt{1/768} = \sqrt{3}$, while $T_{\mathrm{graph}}(S)$ remains zero (since $x - \mathrm{mean}(x) = 0$ component-wise).

We now set γ_1 to all zeros, so $\sigma(\gamma_1(\cdot)) = \frac{1}{2}$, and let ψ_1 project its input onto the coordinate corresponding to $\mathrm{LN}(T_{\mathrm{graph}})_3$ (a two-layer ReLU MLP suffices, since the relevant inputs are nonnegative). Then $X^{(1)}_{uv}(G) = X^{(0)}_{uv}(G) + \frac{1}{2} \cdot \mathrm{LN}(T_{\mathrm{graph}}(G))_3$. Since $A_{vv} = 0$, the diagonal sum readout gives $\sum_v X^{(1)}_{vv}(R) = 16 \cdot \frac{1}{2} \cdot \sqrt{3} = 8\sqrt{3}$ and $\sum_v X^{(1)}_{vv}(S) = 0$. With the final readout MLP set to the identity, $\Phi(R) = 8\sqrt{3} \neq 0 = \Phi(S)$. □

D Deferred implementation details from Section 4

D.1 Figure of Architecture

We provide a picture of the architecture described in Section 4 below.

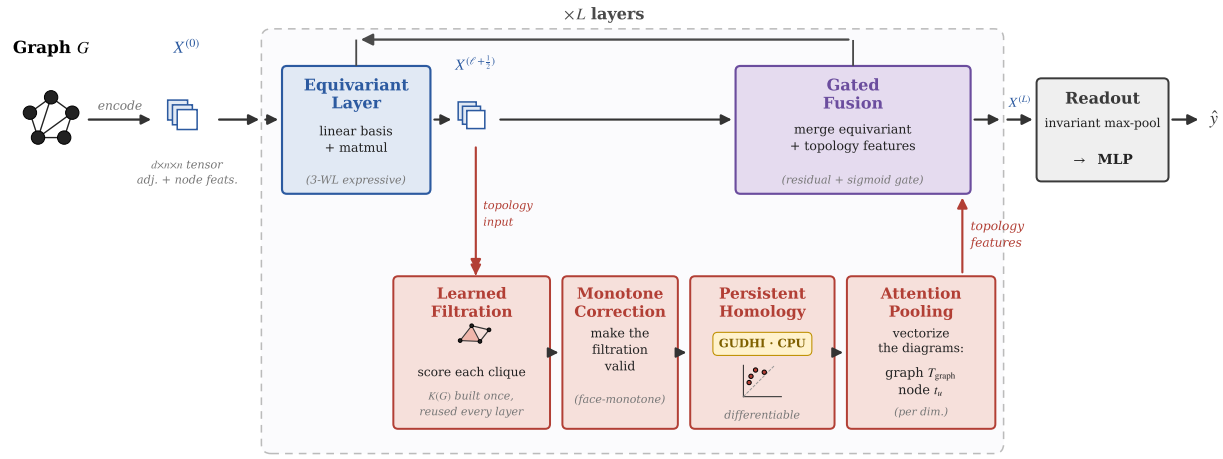


Figure 2: Architecture of the topologically-augmented equivariant network. Each of L layers applies an Equivariant Layer (linear basis of [37] plus a matrix-multiplication branch), feeds the result $X^{(\ell+1/2)}$ through a topology branch (Learned Filtration on $K(G) \to$ Monotone Correction $\to$ Persistent Homology via GUDHI $\to$ per-dimension Attention Pooling), and combines the two streams via a Gated Fusion to produce $X^{(\ell+1)}$. An invariant max-pool readout followed by an MLP produces $\hat{y}$.

D.2 Tie-merging for equivariant node attribution

The definition $\mathcal{N}_i = \sigma_b \cup \sigma_d$ uses the specific birth and death simplices chosen by GUDHI. When two simplices of the same dimension share a filtration value, GUDHI breaks the tie by lexicographic order on vertex label, which is not isomorphism-invariant. This is because relabelling vertices can change which simplex appears as the representative, thus changing $\mathcal{N}_i$ and breaking equivariance. We illustrate this more clearly with an example.

Example 7. *Consider two edges* $(0,1)$ *and* $(2,3)$ *with equal filtration values* 0.37. *Since* $(0,1)$ *is lexicographically smaller,* GUDHI *inserts it first, and suppose it becomes the birth simplex of some persistence pair* i*, so* $\mathcal{N}_i$ *includes the vertices* $\{0,1\}$*. We now apply the permutation* $g = (02)(13)$*. The edges become* $(2,3)$ *and* $(0,1)$*, which are the same as before, but swapped. For equivariance we need* $\mathcal{N}_i^{g\cdot G} = g(\mathcal{N}_i^G) = \{2,3\}$*, but* GUDHI *again selects* $(0,1)$ *as the birth simplex since it is lexicographically first. Thus,* $\mathcal{N}_i \supset \{0,1\} \neq \{2,3\}$ *and equivariance is not satisfied.*

We resolve this by merging all simplices that share the same dimension and filtration value. For each dimension d and value a, let

$$V_{d,a} = \bigcup_{\substack{\sigma \in K(G) \\ \dim(\sigma)=d,\ \tilde{f}(\sigma)=a}} \sigma.$$

For the i-th persistence pair (σ_b, σ_d) with $\dim(\sigma_b) = d_b$, $\tilde{f}(\sigma_b) = a_b$, $\dim(\sigma_d) = d_d$, and $\tilde{f}(\sigma_d) = a_d$, we redefine

$$\mathcal{N}_i = V_{d_b,a_b} \cup V_{d_d,a_d}.$$

We refer to this correction as *tie-merging*. The key property is that the merged involved-nodes sets now transform correctly under permutations.

Proposition 8. *For any permutation $g \in S_n$ and graph G, there exists a bijection ϕ between the persistence pairs of G and those of $g \cdot G$, preserving birth and death filtration values, such that for every pair i in G the involved-nodes sets transform equivariantly:*

$$\mathcal{N}_{\phi(i)}^{g\cdot G} \;=\; g\big(\mathcal{N}_i^G\big).$$

Proof. By Proposition 1, the persistence diagrams of G and $g \cdot G$ coincide as multisets of birth–death values. Thus, we fix any bijection ϕ pairing entries with equal dimensions and equal birth/death values. Under a permutation g, each simplex σ maps to $g \cdot \sigma$, which has the same dimension and filtration value by Lemma 2. Therefore

$$V_{d,a}^{g\cdot G} \;=\; \bigcup_{\substack{\sigma' \in K(g\cdot G)\\ \dim(\sigma')=d,\, \tilde{f}(\sigma')=a}} \sigma' \;=\; \bigcup_{\substack{\sigma \in K(G)\\ \dim(\sigma)=d,\, \tilde{f}(\sigma)=a}} g(\sigma) \;=\; g\big(V_{d,a}^G\big).$$

By construction of ϕ, the pair $\phi(i)$ in $g \cdot G$ has the same birth/death filtration values (a_b, a_d) and dimensions (d_b, d_d) as pair i in G, so

$$\mathcal{N}_{\phi(i)}^{g\cdot G} \;=\; V_{d_b,a_b}^{g\cdot G} \cup V_{d_d,a_d}^{g\cdot G} \;=\; g\big(V_{d_b,a_b}^G\big) \cup g\big(V_{d_d,a_d}^G\big) \;=\; g\big(\mathcal{N}_i^G\big),$$

as desired. □

With this proposition in place, we can establish the equivariance of the node-level features defined in Section 4.3.

Proposition 9. *For any permutation $g \in S_n$, graph G, and vertex $u \in V$, $t_{g(u)}^{g\cdot G} = t_u^G$.*

Proof. Let ϕ be the bijection of Proposition 8. By construction, ϕ preserves birth and death filtration values, so corresponding pairs have the same lifetimes $\ell_{\phi(i)} = \ell_i$, and consequently $e_{\phi(i)} = e_i$ and $\alpha_{\phi(i)} = \alpha_i$. By Proposition 8, $g(u) \in \mathcal{N}_{\phi(i)}^{g\cdot G}$ if and only if $u \in \mathcal{N}_i^G$. Therefore,

$$t_{g(u)}^{g\cdot G} = \sum_{\phi(i):\, g(u)\in\mathcal{N}_{\phi(i)}^{g\cdot G}} \frac{\exp\big(\alpha_{\phi(i)}\big)}{\sum_{\phi(j):\, g(u)\in\mathcal{N}_{\phi(j)}^{g\cdot G}} \exp\big(\alpha_{\phi(j)}\big)}\, e_{\phi(i)} = \sum_{i:\, u\in\mathcal{N}_i^G} \frac{\exp(\alpha_i)}{\sum_{j:\, u\in\mathcal{N}_j^G} \exp(\alpha_j)}\, e_i = t_u^G,$$

which is the desired result. □

In practice, ties are rare after a few training steps since the learned filtration MLP generically produces distinct values. However, ties can occur at initialization on graphs without node labels, where structurally equivalent vertices receive identical filtration values. In these cases, tie-merging is necessary for correctness.

D.3 Proof of Lemma 4 (Preservation of Permutation Equivariance)

Proof. We need to show that $X_{g(u),g(v)}^{(\ell+1)}$ in $g \cdot G$ equals $X_{uv}^{(\ell+1)}$ for all graphs G and $g \in S_n$.

The input to the fusion has three components. The tensor $X^{(\ell+\frac{1}{2})}$ is equivariant by our construction in Section 3, so $X_{g(u),g(v)}^{(\ell+\frac{1}{2}),g\cdot G} = X_{uv}^{(\ell+\frac{1}{2}),G}$. The graph-level vector T_{graph} is invariant by Proposition 1, so $T_{\text{graph}}^{g\cdot G} = T_{\text{graph}}^G$. The node-level features are equivariant by Proposition 9, so

$t^{g\cdot G}_{g(u)} = t^G_u$ and $t^{g\cdot G}_{g(v)} = t^G_v$. LayerNorm normalizes across the feature dimension at each position independently and does not mix positions, so it preserves these properties. Combining these, the concatenation satisfies

$$[X^{(\ell+\frac{1}{2}),\, g\cdot G}_{g(u),g(v)} \parallel T^{g\cdot G}_{\text{graph}} \parallel t^{g\cdot G}_{g(u)} \parallel t^{g\cdot G}_{g(v)}] = [X^{(\ell+\frac{1}{2}),\, G}_{uv} \parallel T^{G}_{\text{graph}} \parallel t^{G}_{u} \parallel t^{G}_{v}].$$

Since γ_ℓ and ψ_ℓ are pointwise MLPs with shared parameters applied independently at each position, they output the same value at $(g(u), g(v))$ in $g \cdot G$ as at (u, v) in G, since their inputs are equal by the above. The sigmoid σ, Hadamard product $\odot$, and addition are all elementwise, so they preserve this equality. Therefore $X^{(\ell+1),g\cdot G}_{g(u),g(v)} = X^{(\ell+1),G}_{uv}$. □

E Diagnostic Analysis: When Does The Model Use Topology

This appendix provides the full diagnostic analysis summarized in Section 5.3. The TU benchmark results in Table 1 show a non-uniform pattern: topology yields consistent gains on the four molecular datasets but is roughly neutral on IMDB-B and slightly worsens performance on PROTEINS. A natural question is whether this pattern is reflected in the model's internal behavior, that is, whether the model itself learns when topology is informative and modulates its use of the topology branch accordingly.

We answer this affirmatively by instrumenting the trained models with three families of diagnostics: (i) the per-layer gate activation $\sigma(\gamma_\ell)$ from the gated residual fusion in Section 4.4, recorded every epoch; (ii) the average number of finite persistence pairs per graph and the empirical mean and standard deviation of filtration values, at the start and end of training; and (iii) the fraction of persistence pairs modified by tie-merging (Appendix D.2). Across MUTAG, PTC, NCI1, IMDB-B, and PROTEINS, the deep-layer gate exhibits three qualitatively distinct regimes of behavior at convergence, and these regimes correspond directly to the accuracy outcome on each dataset.

Setup. For each dataset we run a single training pass with one held-out validation split (the same train/validation partition used in one fold of the 7-fold CV protocol of Section 5.1), matching architecture and topology-branch hyperparameters across runs (block features $[64, 64]$, MLP depth 2, max_ph_dim $= 1$, max_simplex_dim $= 2$, hidden_dim $= 16$). The training and validation curves shown in Figure 8 and the test-time per-batch gate distribution shown in Figure 6(f) refer to this held-out partition.

The diagnostic statistics throughout this appendix are computed for the v2 (birth–persistence) variant of Section 5, which is the headline method on every dataset where topology contributes; the ΔAcc column in Table 3 reports v2's gain over the baseline.

Gate behavior. Figure 3 shows the gate trajectory at both layers across training, and Figure 4 plots the deep-layer gate's last-50-epoch mean against its last-50-epoch standard deviation. Two observations stand out.

First, the shallow gate is largely uninformative. Across all four datasets it remains close to its initialization value of $\sigma(2.0) \approx 0.88$, with only modest drift. The deep gate, by contrast, separates the datasets cleanly. This asymmetry is unsurprising, because the shallow layer's filtration MLP operates on the raw input pairwise tensor (essentially the adjacency matrix and node attributes), whereas the deep layer's filtration MLP operates on features (which have already been equivariantly transformed) that carry dataset-specific structure, so it is the deep gate that has the discriminative material to respond to.

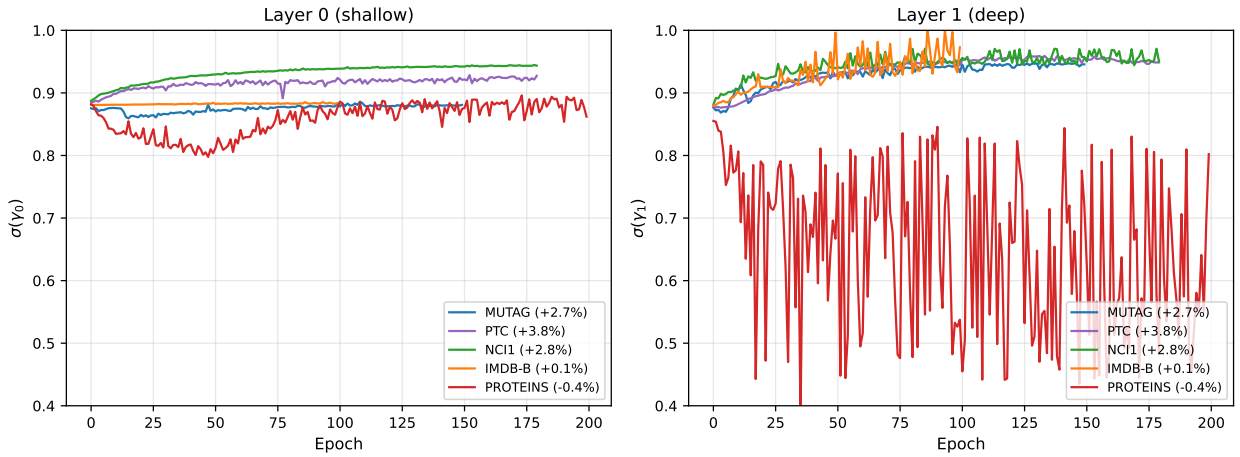


Figure 3: Gate activation $\sigma(\gamma_\ell)$ at both layers over training, for the five diagnostic datasets. The shallow gate (left) remains close to initialization across all datasets. The deep gate (right) separates them into three regimes that correspond to the test accuracy outcome shown in the legend.

Table 3: Diagnostic statistics on the five datasets. Gate mean and standard deviation are over the last 50 epochs. Filtration std is the empirical spread of layer-1 filtration values at the end of training. The regime is determined by joint position in the (mean, std) plane.

Dataset	ΔAcc	$\overline{\sigma(\gamma_1)}$	std($\sigma(\gamma_1)$)	filt. std (final)	$H_0 + H_1$ pairs (final)	Regime
MUTAG	+2.7%	0.944	0.0036	0.39	10.6	stable commitment
PTC	+3.8%	0.953	0.0033	0.40	3.0	stable commitment
NCI1	+2.8%	0.956	0.0083	0.27	12.9	stable commitment
IMDB-B	+0.1%	0.951	0.0221	0.013	351.5	noisy commitment
PROTEINS	−0.4%	0.608	0.1175	22.29	3.1	indecision

Second, the deep gate exhibits three regimes distinguished by its convergence behavior:

Stable commitment, a high and steady gate (mean ≥ 0.85, std $< 1.5 \times 10^{-2}$), observed on MUTAG, PTC, and NCI1. The gate converges within ≈ 50 epochs to $\sigma(\gamma_1) \approx 0.95$ and remains there with standard deviation below 1.5×10^{-2} over the final 50 epochs. The model commits to using almost the full topological signal, and accuracy improves by +2.7%, +3.8%, and +2.8% respectively.

Noisy commitment, a high but unstable gate (mean ≥ 0.85, std in $[1.5 \times 10^{-2}, 10^{-1}]$), observed on IMDB-B. The gate's mean is comparable to the molecular case (≈ 0.95), but its standard deviation is roughly seven times larger (2.2×10^{-2}), and individual epochs visit values as high as 1.00 and as low as 0.92. The model wants to use topology but cannot find a stable parameterization. Accuracy is essentially unchanged (+0.1%).

Indecision, a low and oscillating gate (mean < 0.85), observed on PROTEINS. The deep gate drops to mean 0.61 with standard deviation 0.12, oscillating between 0.44 and 0.83 throughout training. The model is pulled in conflicting directions and never commits to either using or ignoring topology. Accuracy regresses by 0.4%.

Mechanism. The topological diagnostics make the underlying mechanisms behind the three regimes transparent.

For MUTAG, PTC, and NCI1, the filtration MLP develops a moderate but well-controlled spread of values, with the layer-1 filtration std growing by roughly 2 to 4× over training (from 0.07 to 0.19

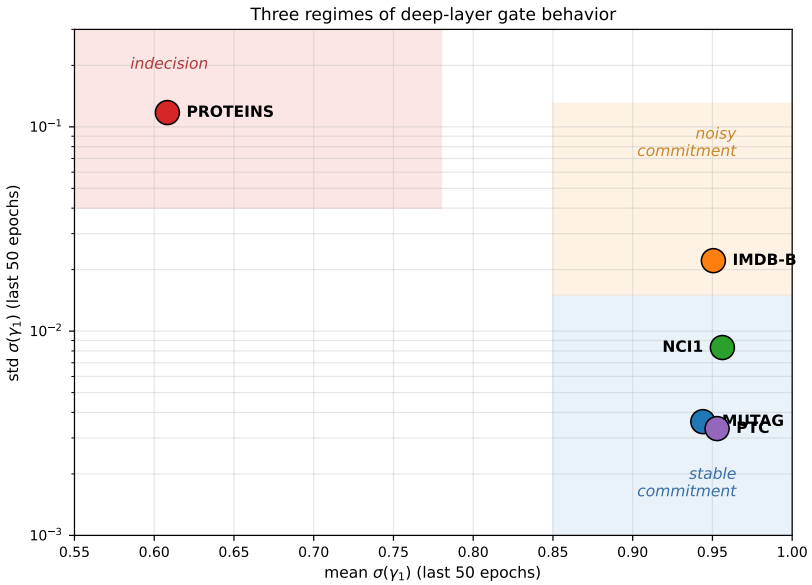


Figure 4: Mean versus standard deviation of the deep-layer gate $\sigma(\gamma_1)$ over the last 50 epochs of training. Each dataset occupies one of three regions: stable commitment (high mean, low std), noisy commitment (high mean, moderate std), and indecision (low mean, high std). The regime corresponds to the accuracy outcome of Table 1.

at initialization, ending at 0.27 to 0.40). This produces meaningful sub-level filtrations and stable persistence diagrams. Topology becomes a useful additional channel and the gate commits to it. NCI1 sits at the looser end of this group, with a final filtration std of 0.27 and a slightly larger gate variance (8.3×10^{-3}) than MUTAG or PTC, but the qualitative pattern is the same and the accuracy gain (+2.8%) is comparable to MUTAG.

For IMDB-B, the structural diagnostics reveal a more nuanced story. The layer-0 filtration is degenerate: filtration values have empirical std $\approx 10^{-4}$ both at initialization and after training, and the corresponding clique complex produces zero finite persistence pairs at any point during training. This is the predictable consequence of a featureless graph; with no node attributes the input pairwise tensor reduces to the adjacency matrix, and a shared MLP applied to such homogeneous inputs cannot break the symmetry. By layer 1, the equivariant network has produced enough feature variation that the filtration MLP can do something nontrivial, generating an average of 40 H_0 pairs and 312 H_1 pairs per graph at convergence. However the empirical filtration spread remains very small (0.013), so the topological signal is qualitatively weak, and the gate cannot find a value it is confident in. Hence the noisy commitment regime.

For PROTEINS, the failure mode is different. Here the layer-1 filtration spread blows up by nearly 30× during training, from 0.75 to 22.29, with the empirical mean drifting from -1.7 to $+42$. The number of finite persistence pairs collapses (H_0 from 37 to 3, H_1 from 15 to 0.3), since extreme filtration values push almost all features into the essential regime that we discard. The filtration MLP is no longer producing a useful sub-level filtration; it is producing a pathological one. The deep gate responds by partially disengaging, and its inability to settle on a single value reflects the optimization pulling between conflicting signals from this degenerate filtration.

Across all four datasets, the picture is consistent: the deep gate's behavior at convergence is a direct readout of whether the filtration MLP has settled into an informative regime, and the test accuracy follows the filtration quality.

Tie-merging. The tie-merging correction described in Appendix D.2 is required for theoretical equivariance whenever distinct simplices receive the same filtration value. Across all five datasets and 100 to 200 epochs of training, the empirical fraction of persistence pairs whose involved-nodes set is modified by tie-merging is below 1%, with the single exception of MUTAG layer 0 at the final epoch (7.8%). NCI1 holds tie fraction at 0 throughout the run on both layers. The learned filtration MLP generically produces distinct values, and the practical contribution of tie-merging is bounded; it functions as a theoretical safeguard rather than a frequent operational correction. This validates the analysis in Appendix D.2.

Discussion. The diagnostic analysis paints a coherent picture of how the architecture uses topology in practice. The gated residual fusion is not a passive pass-through; the deep-layer gate is responsive to the quality of the filtration that the filtration MLP produces, and its behavior at convergence is a direct empirical signal of when topology is genuinely useful. The same datasets where topology yields the strongest accuracy gains (MUTAG, PTC, NCI1) are the datasets on which the model commits most confidently to using it, and the dataset where topology is harmful (PROTEINS) is the one on which the gate refuses to commit. IMDB-B sits between these extremes in both senses. Its filtration is mechanically constrained by the absence of node features, the topology branch produces a high-volume but low-spread signal, and accordingly the gate stays high on average while failing to stabilize.

This is consistent with the broader theoretical claim that topology helps when filtrations are informative. Rather than being asserted, the correspondence is here observed directly: filtration quality, gate behavior, and accuracy outcome track each other on a per-dataset basis.

Additional plots. Figure 6 collects the gate dynamics, regime scatter, filtration spread, persistence pair counts, and test-time gate distribution into a single panel for comparison. Figures 5 and 7 present the two key topological diagnostics in isolation, and Figure 8 shows the training and validation loss and accuracy across all four datasets.

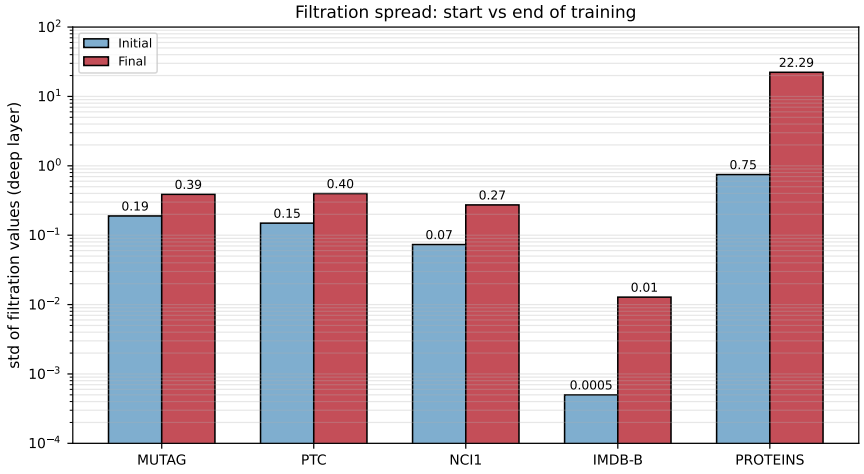


Figure 5: Standard deviation of the layer-1 filtration values at the start and end of training, on a logarithmic scale. The filtration MLP develops a moderate spread on MUTAG, PTC, and NCI1, remains essentially flat on IMDB-B due to the absence of node features, and explodes by nearly 30× on PROTEINS, indicating a pathological filtration that the deep gate refuses to commit to.

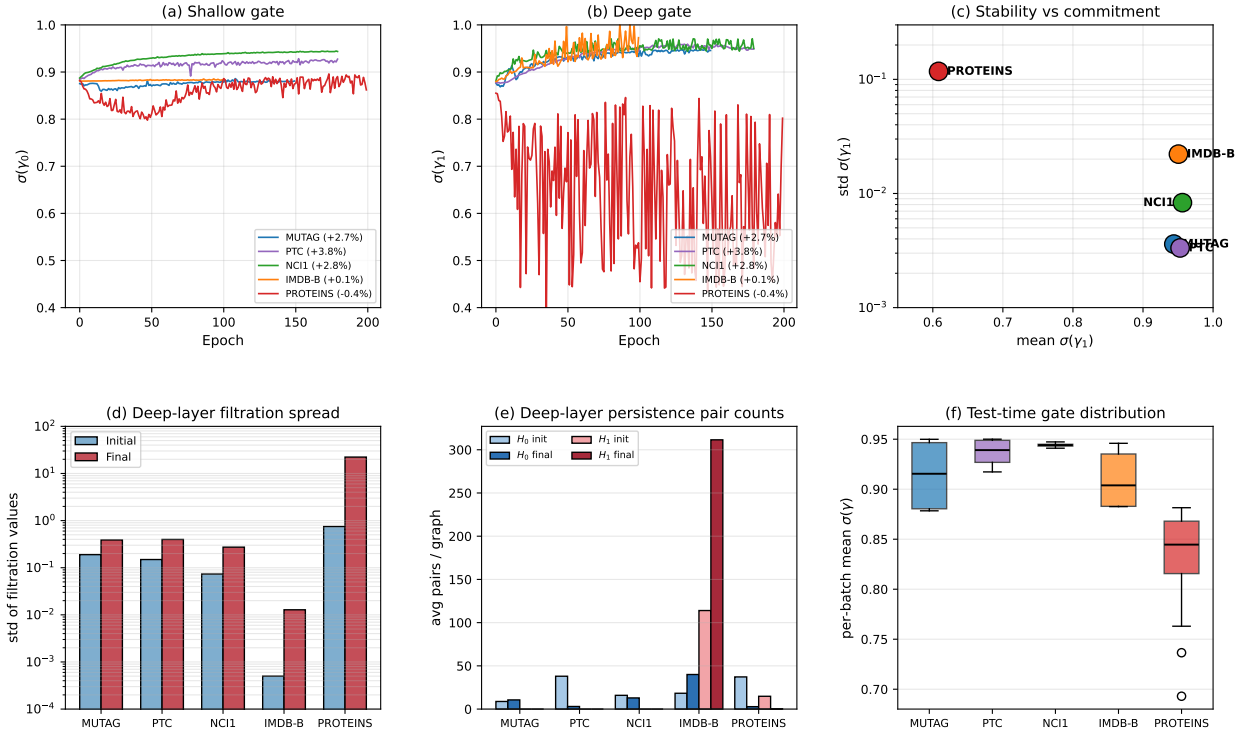


Figure 6: Comprehensive diagnostic dashboard for the five datasets. **(a, b)** Gate activation $\sigma(\gamma_\ell)$ at the shallow and deep layers over training. **(c)** Mean versus standard deviation of the deep-layer gate over the last 50 epochs, showing the three regimes. **(d)** Standard deviation of the layer-1 filtration values at initialization and at the end of training, on a logarithmic scale. MUTAG, PTC, and NCI1 develop a moderate spread, IMDB-B remains nearly flat, and PROTEINS exhibits a 30× blow-up to a pathological regime. **(e)** Average number of finite persistence pairs per graph at the deep layer. IMDB-B's layer-1 features generate a large H_1 count from social-network structure, while PTC and PROTEINS pair counts collapse during training. MUTAG and NCI1 stay roughly constant. **(f)** Test-time per-batch gate distribution (boxplot). The training-time pattern is preserved on held-out batches.

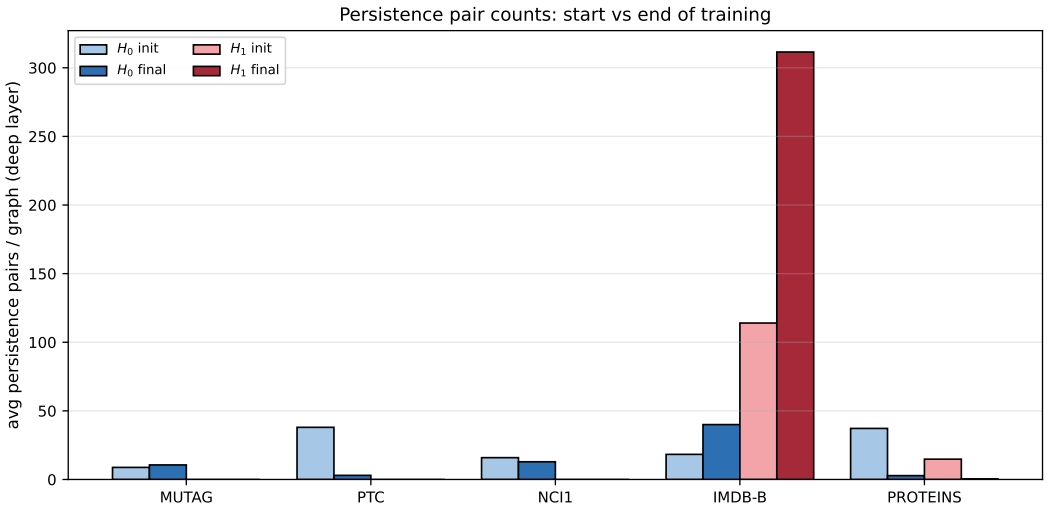


Figure 7: Average number of finite persistence pairs per graph at the deep layer, at initialization and at the end of training. Several patterns are worth noting. MUTAG and NCI1 remain stable, PTC's H_0 count collapses as the filtration concentrates, IMDB-B generates a very large H_1 population reflecting the cycle-rich structure of social networks, and PROTEINS's pair counts collapse together with the pathological filtration spread shown in Figure 5.

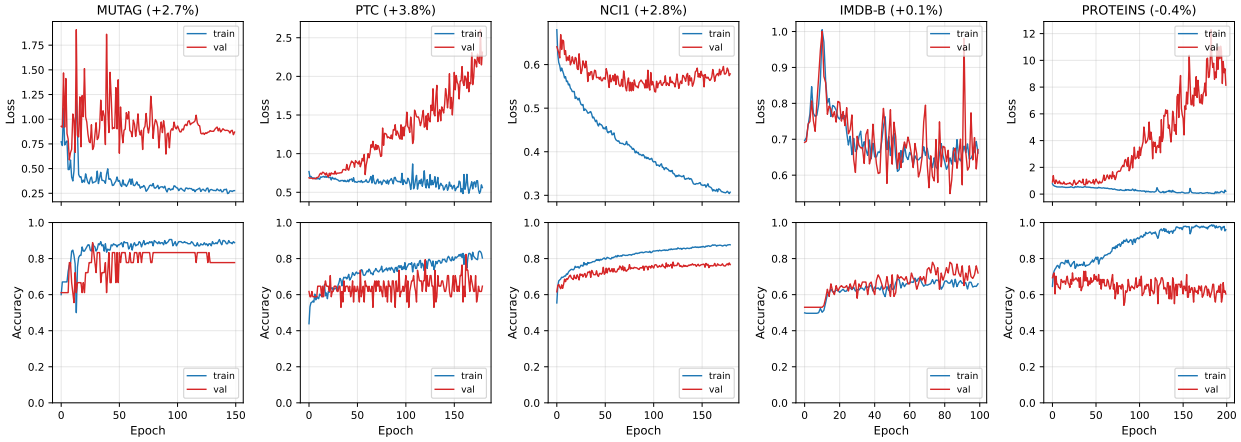


Figure 8: Training and validation loss and accuracy on the five diagnostic runs. MUTAG, NCI1, and IMDB-B train cleanly, with validation curves tracking training. PTC and PROTEINS exhibit progressive validation-loss divergence, most pronounced on PROTEINS where validation loss climbs to ≈ 10 while training loss approaches zero. This overfitting is consistent with the pathological filtration regime documented in Appendix E. When the topology branch produces an unstable signal, the model has no useful gradient direction to commit to and instead memorizes noise.

F Richer Topological Representations

The base architecture of Section 4 uses a minimal topological representation, since it only considers scalar lifetimes, with essential features discarded. In this appendix, we describe a variant that extracts additional information from each persistence pair. All variants share the same filtration, simplicial complex, gated residual fusion, and tie-merging mechanism.

F.1 Birth–persistence embedding

The base architecture discards the birth time of each persistence pair, embedding only the lifetime $\ell_i = \tilde{f}(\sigma_d) - \tilde{f}(\sigma_b)$. However, the position of a pair in the birth–persistence plane carries task-relevant information, since two pairs with the same lifetime but different birth times correspond to topological features that appear at different scales of the filtration. This observation is crucial to the WKPI framework of [58], which learns a weighted kernel over the birth–persistence plane and demonstrates that the two-dimensional representation captures structure invisible to lifetime alone.

We replace the scalar lifetime input with the two-dimensional vector $\mathbf{p}_i = (\tilde{f}(\sigma_b), \ell_i) \in \mathbb{R}^2$ and modify the embedding and attention networks to accept $\mathbb{R}^2$ as inputs: $\mathrm{MLP}^{(p)}_{\mathrm{embed}}(\mathbf{p}_i) \in \mathbb{R}^{d_v}$ and $\alpha_i = \mathrm{MLP}^{(p)}_{\mathrm{attn}}(\mathbf{p}_i) \in \mathbb{R}$. Before embedding, we normalize each $\mathbf{p}_i$ by the per-graph, per-homological-dimension mean $\mu^{(p)}$ and population standard deviation $\sigma^{(p)}$:

$$\hat{\mathbf{p}}_i \;=\; \frac{\mathbf{p}_i - \mu^{(p)}}{\sigma^{(p)} + \epsilon},$$

where $\epsilon > 0$ is a small constant that handles the degenerate case in which all pairs within a homological dimension share identical birth–persistence coordinates (so $\sigma^{(p)} = 0$). Since the normalization statistics are computed over the unordered multiset $\{\mathbf{p}_i\}$, the normalization is permutation-invariant and preserves the equivariance properties of Section 3. The multiset $\{\mathbf{p}_i\}$ is invariant under permutations by Proposition 1, and the normalization statistics only depend upon the multiset $\{\mathbf{p}_i\}$, not the ordering of their pairs, so they are also invariant. All subsequent operations are identical to the base architecture, so all equivariance and invariance properties are preserved, including Proposition 9 for node-level features.

The multiset $\{\mathbf{p}_i\}$ is invariant under permutations by Proposition 1, and the normalization statistics depend only on this multiset rather than on its ordering, so they too are invariant. All subsequent operations are identical to the base architecture, so all equivariance and invariance properties are preserved, including Proposition 9 for node-level features.

This variant captures additional information about the *relative* position of persistence pairs within the birth–persistence plane: two diagrams with identical lifetime distributions but different birth-time distributions yield different normalized inputs and hence different embeddings. We note that per-graph normalization removes absolute scale information from each graph's persistence diagram, so the variant is not strictly information-preserving over the base architecture in the absolute sense; however, the discriminative power gained by access to the second coordinate is sufficient to preserve the strict expressivity gain over 3-WL, as we now establish.

F.2 Expressivity

We conclude by analyzing the expressivity of both the variants.

Corollary 10. *The birth–persistence variant is strictly more expressive than* 3*-WL.*

Proof. The lower bound (at least k-WL expressive) follows by the same argument as Theorem 3: setting the gate $\sigma(\gamma_\ell) \to 0$ recovers the pure equivariant network of [36]. We now exhibit explicit parameters that separate the Rook–Shrikhande pair, adapting the construction of Corollary 6 to account for the normalization step.

We now take the same setting as in the proof of Corollary 6: $L = 1$, $D = 3$, $\mathrm{EqvLayer}_1 = \mathrm{Id}$, $\rho_1 = \mathrm{Id}$, so that $K(R)$ has eight H_2 pairs at $(2/3, 3/4)$ with lifetime $1/12$ and $K(S)$ has none. The normalized birth–persistence inputs for R satisfy $\hat{\mathbf{p}}_i = \mathbf{0}$ for all eight pairs, since the multiset is constant in both coordinates and $\mu^{(2)} = (2/3, 1/12)$, $\sigma^{(2)} = (0, 0)$, so $\hat{\mathbf{p}}_i = (\mathbf{p}_i - \mu^{(2)})/(0 + \epsilon) = \mathbf{0}$.

Set $\mathrm{MLP}^{(2)}_{\mathrm{embed}}$ to be the constant map $\mathbf{x} \mapsto 1$ (a two-layer ReLU MLP with zero first-layer weights and unit bias on the output, which is realizable). Then $e_i = 1$ for each of the eight pairs of R, and $T^{(2)}_{\mathrm{graph}}(R) = 1$ under the uniform-softmax convention $\mathrm{MLP}^{(2)}_{\mathrm{attn}} \equiv 0$. For S, $T^{(2)}_{\mathrm{graph}}(S) = 0$ by the convention that empty persistence dimensions contribute zero (Section 4.2). For the remaining homological dimensions, set $\mathrm{MLP}^{(p)}_{\mathrm{attn}}, \mathrm{MLP}^{(p)}_{\mathrm{embed}} \equiv 0$ for $p \in \{0, 1, 3\}$. With these choices, $T_{\mathrm{graph}}(R) = (0, 0, 1, 0)$ and $T_{\mathrm{graph}}(S) = \mathbf{0} \in \mathbb{R}^4$.

After LayerNorm, the third coordinate of $\mathrm{LN}(T_{\mathrm{graph}}(R))$ equals $(1 - \frac{1}{4})/\sqrt{3/16} = \sqrt{3}$, while $\mathrm{LN}(T_{\mathrm{graph}}(S)) = \mathbf{0}$ (a zero vector remains zero under LayerNorm with the standard ϵ-stabilization). The remainder of the construction proceeds identically to the proof of Corollary 6: setting $\gamma_1 \equiv 0$ and ψ_1 to project onto $\mathrm{LN}(T_{\mathrm{graph}})_3$ yields $\Phi(R) = 8\sqrt{3} \neq 0 = \Phi(S)$. □

G Computational Complexity

We analyze the computational cost of the topology branch and discuss the implications for scalability relative to the equivariant baseline.

Asymptotic bounds. Let $n = |V|$ be the number of vertices and D the maximum simplex dimension at which the clique complex is constructed (Section 4). The number of simplices in $K(G)$ is bounded by

$$N \;\leq\; \sum_{k=0}^{D} \binom{n}{k+1} \;=\; O(n^{D+1}),$$

with equality realized on a complete graph. For a sparse graph with m edges, the count is bounded more tightly by $\sum_{k=0}^{D} c_k(G)$, where $c_k(G)$ is the number of $(k+1)$-cliques. On TU benchmark graphs this is typically far below the worst-case bound.

Persistent homology computation via boundary-matrix reduction [22] has worst-case complexity $O(N^3)$ in the number of simplices [38], yielding a worst-case bound of $O(n^{3(D+1)})$ for the full topology branch. This bound is sharp in adversarial constructions [38], but the empirical behavior on realistic data is much closer to linear, as the boundary matrix tends to remain sparse during reduction, and recent analyzes have given tighter expected-case bounds for clique-style filtrations on random graphs [4]. On the TU benchmarks considered here ($n \leq 620$, $D \leq 2$), reductions complete in milliseconds per graph.

Per-layer cost. The forward pass at a single layer has three steps. The first is constructing the simplicial complex $K(G)$ from the input adjacency, which is performed once per graph and amortized across layers, since $K(G)$ depends only on the fixed edge set of G. The second is evaluating the filtration MLP ρ_ℓ on each of the N simplices, costing $O(Nd^2)$ for a hidden dimension d MLP. The third and final step is reducing the boundary matrix to obtain persistence pairs. In the worst case

this is $O(N^3)$, however, it is near-linear in practice. The vectorization step (Section 4.2) adds $O(Nd_v)$ for the per-pair embeddings and softmax pooling.

GPU–CPU transfer. GUDHI executes on CPU, while the equivariant network runs on GPU. At each forward pass through the topology branch, the per-simplex filtration values produced by ρ_ℓ are transferred from GPU to CPU for boundary-matrix reduction, and the resulting persistence-pair indices (which simplices pair with which) are transferred back to GPU. The transferred payload is $O(N)$ scalars in each direction. The backward pass does not require an additional round trip: the persistence pairing is treated as fixed (locally constant on a measure-one set; see Section 4.1), and the differentiable lifetimes $\ell_i = \max(\tilde{f}(\sigma_d) - \tilde{f}(\sigma_b), 0)$ are computed on GPU through standard subtraction operations, so gradient flow through autograd remains GPU-resident. The per-layer overhead is therefore dominated by the fixed synchronization cost of the forward transfer rather than by the data volume itself.

Birth–persistence variant. The birth–persistence variant of Appendix F shares the simplicial complex construction and persistent homology computation with the base architecture, so the dominant cost terms are unchanged. The only modifications are (i) doubling the input dimension of $\mathrm{MLP}^{(p)}_{\mathrm{embed}}$ and $\mathrm{MLP}^{(p)}_{\mathrm{attn}}$ from 1 to 2, which increases the per-pair embedding cost by a small constant factor, and (ii) the per-graph normalization step, which adds an $O(N_p)$ pass per homological dimension to compute mean and standard deviation. Both contributions are negligible relative to the boundary-matrix reduction and equivariant-layer costs. The GPU–CPU transfer pattern is identical to the base architecture.

Scalability. Two consequences follow. First, in the regime considered in this paper ($n \leq 620$, $D \leq 2$), the topology branch is not the bottleneck of training: GPU–CPU synchronization dominates the per-layer overhead, but the absolute cost remains small because the boundary-matrix reduction is fast in this regime. Second, scalability degrades for graphs with n in the thousands or for $D \geq 3$, both because the simplex count grows as $O(n^{D+1})$ and because per-graph GPU–CPU transfers cannot be batched as effectively as GPU-resident operations. Scaling to OGB-sized graphs would benefit from a GPU-resident persistence implementation; we view this as an important avenue for future engineering work but outside the scope of the present paper.