Homological Representation Learning for Molecular Graphs*

Yoshihiro Maruyama MARUYAMA@I.NAGOYA-U.AC.JP
School of Informatics, Nagoya University, Nagoya, Japan
Arisa Yasuda ARISA.YASUDA@ANU.EDU.AU
School of Computing, Australian National University, Canberra, Australia

Editors: List of editors' names

Abstract

We propose *Homological Representation Learning* (HomRL), an architecture-agnostic regularization for graph encoders that aligns latent embeddings with an efficiently computable homological signature of the input. Concretely, alongside the task head we train a linear *homological head* to regress $s(G) = \big(\beta_1(G),\ \chi(G),\ |V(G^{(2)})|/|V(G)|,\ \#\mathrm{comp}(G^{(2)})\big)$, where β_1 is the cycle rank (aka. the first Betti number), χ is the Euler characteristic, $G^{(2)}$ is the 2-core, and #comp is the number of connected components. These quantities are additive (except for the normalized ratio) and computable in $O(|V|+|E|)$ time. Theoretically, if the homological head fits with error at most ε and two graphs satisfy $s(G)=s(G')$, then their embeddings obey $\|z_\theta(G) - z_\theta(G')\| \le 2\varepsilon/\sigma_{\min}(W_m)$, and more generally with $\delta = \|s(G) - s(G')\|$ we have $\|z_\theta(G) - z_\theta(G')\| \le (2\varepsilon+\delta)/\sigma_{\min}(W_m)$. Empirically, HomRL significantly improves accuracy in a molecular graph task with identical encoders, optimizers, and budgets. HomRL, with negligible computational overhead, provides interpretable supervision, and complements symmetry-aware architectures.

Keywords: representation learning; homology; algebraic topology; molecular graph

1. Introduction

Deep encoders for discrete structures such as molecular graphs must remain reliable under transformations that preserve the underlying scaffold (for example, attaching pendant trees, rewiring branches away from ring systems, or modest attribute corruption). Standard objectives encourage task fit but do not explicitly control how representations respond to such scaffold-preserving edits. This paper introduces *Homological Representation Learning* (HomRL), a principled addition to encoders that aligns learned features with computable homological invariants of the input, thereby injecting additivity and cycle-space awareness into the representation.

Given a finite graph $G = (V, E)$ with node features X, let $z_\theta(G) \in \mathbb{R}^d$ be the output of a permutation-invariant encoder (for example, message passing + pooling) (Gilmer et al., 2017; Kipf and Welling, 2017; Hamilton et al., 2017). Many scaffold-preserving edits leave global topological information unchanged: the cycle rank $\beta_1(G)$ (aka. the first Betti number), the Euler characteristic χ, and statistics of the 2-core $G^{(2)}$ obtained by iteratively pruning degree-one vertices (Seidman, 1983; Batagelj and Zaveršnik, 2003; Matula and Beck, 1983). We collect these into a *homological signature*

$$s(G) \;=\; \Big(\beta_1(G),\ \chi(G),\ \tfrac{|V(G^{(2)})|}{|V(G)|},\ \#\mathrm{comp}(G^{(2)})\Big) \in \mathbb{R}^4,$$

* This work was supported by JST Moonshot R&D JPMJMS2033, JST PRESTO JPMJFR206P, JST FOREST JPMJPR24K9, and JSPS KAKEN 25K03181.

computable homological information that is additive on disjoint unions and invariant to many tree-like edits.[1] Alongside the usual classification head $\hat{y}$, we train a *homological head* $\widehat{s} = W_m z_\theta(G) + b_m$ to regress $s(G)$. The training loss augments the task objective with an ℓ_2 penalty:

$$\mathcal{L} \;=\; \mathrm{CE}(\hat{y}, y) \;+\; \lambda\, \|\widehat{s} - s(G)\|_2^2 \quad (\text{optionally } +\mu\, \|\widehat{s}(a{\cdot}G) - \widehat{s}(G)\|_2^2 \text{ for augmentations } a).$$

This encourages z_θ to factor through an additive, deformation-stable summary of G. In particular, when W_m is well conditioned and the regression error is small, scaffold-preserving edits that leave $s(G)$ fixed induce small changes in $z_\theta(G)$.

Why does this help? (i) *Inductive bias:* homological targets inject cycle-space information that ordinary task losses do not enforce. (ii) *Additivity and stability:* the signature $s(\cdot)$ is additive over disjoint unions and invariant to pendant-tree moves; matching it constrains representation drift across such edits. (iii) *Compatibility:* HomRL is a drop-in head and loss; it leaves the encoder class, capacity, and optimizer unchanged.

The contributions of this paper can be summarized as follows:

- **Formalism.** We formulate HomRL as learning a representation that approximately factors through an additive, homology-derived signature $s(\cdot)$.
- **Theory.** Under a mild assumption on W_m, if two graphs share the same signature and the homological head fits with error at most ε, their embeddings satisfy a stability bound of the form $\|z_\theta(G) - z_\theta(G')\| \le c\,\varepsilon$ for a constant c depending on W_m.
- **Experiment.** With an identical encoder for both methods, HomRL improves binary ring-system classification over a baseline by +4.83 points on the test set (from 0.6517 to 0.7000) and by +5.50 points under an out-of-distribution shift that preserves β_1 (from 0.6317 to 0.6867); the homological regression attains test MSE 0.0905.
- **Practicality.** The signature uses quantities computable in linear or near-linear time in $|V| + |E|$, being computationally tractable.

Note that we deliberately use a tractable signature rather than full homological information; this makes the approach easy to implement but coarse. Nevertheless, the homological constraints already yield measurable robustness gains without increasing model capacity.

1.1. Related Work

Here we compare our method with related work.

1.1.1. Invariant and Equivariant Representation Learning

Group-equivariant neural networks enforce symmetries by architectural design, yielding representations that transform predictably under prescribed actions. Early examples include group-equivariant convolutions for planar images (Cohen and Welling, 2016); broader unifying perspectives for grids, groups, graphs, and gauges are provided by geometric deep learning surveys (Bronstein et al., 2021). On graphs, permutation invariance/equivariance is

1. Graphs are 1-dim. CW complexes, so $\chi(G) = |V| - |E|$. #comp is the number of connected components.

achieved via message passing and set/graph pooling, with expressivity often analyzed through Weisfeiler–Lehman tests and extensions; recent work studies topological expressiveness explicitly (Maron et al., 2019). Our method is complementary: instead of hard-coding a symmetry group, we *regularize* the learned representation to regress additive, homology-derived signatures that remain stable across scaffold-preserving edits.

1.1.2. Topological Data Analysis in Learning

Topological Data Analysis (TDA) provides deformation-invariant summaries (e.g., Betti numbers, persistence diagrams) that capture global structure while being stable to small perturbations; see Carlsson (2009) for an overview. Numerous pipelines inject persistent features into learning via fixed summaries (such as persistence landscapes or images) or differentiable surrogates (Carrière et al., 2017; Adams et al., 2017; Carrière et al., 2020; Hofer et al., 2019). In contrast, we target a compact, task-agnostic signature that is (i) additive under cut-and-paste, (ii) efficiently computable on graphs (cycle rank and 2-core statistics), and (iii) used *as a training target* so that the encoder's intermediate representation aligns with homological structure rather than merely consuming topological features as inputs.[2]

1.1.3. Sheaf-Theoretic Learning on Graphs

Sheaf-theoretic models generalize classical signal processing on graphs by encoding local consistency constraints over cells and computing cohomological invariants (e.g., sheaf Laplacians, cohomology ranks). Overviews of sheaves and cosheaves and their applications are given in Curry (2013). While sheaf neural networks design architectures whose operators are derived from a chosen sheaf, our approach is orthogonal: we keep a standard encoder and impose a *homological head* that regresses global invariants (cycle-space and 2-core summaries), thereby encouraging the latent space to factor through an additive, topology-aware summary without committing to a specific sheaf construction (Hansen and Ghrist, 2019).

1.1.4. Additive Measures and Computable Specializations

Additivity under disjoint unions and compatibility with gluing operations are central in Grothendieck-style constructions: objects are mapped to an abelian (semi)ring where invariants respect cut-and-paste identities. For graphs viewed as 1D CW complexes, the first Betti number β_1 (cycle rank) and Euler characteristic χ already furnish such additive invariants; simple statistics of the 2-core further summarize cycle support. We treat the vector $\Big(\beta_1(G),\ \chi(G),\ \frac{|V(G^{(2)})|}{|V(G)|},\ \#\mathrm{comp}(G^{(2)})\Big)$ as computable homological information for training. Unlike pipelines that *use* a topological descriptor as an extra feature, we *learn* a representation whose linear readout predicts this additive summary, thus encouraging factorization through an invariant of the cut-and-paste semiring of graphs (Bittner, 2004).

1.1.5. Molecular Graphs and Ring-Aware Modeling

In molecular property prediction, ring systems and scaffold structure are known to drive generalization behavior; scaffold-preserving distribution shifts stress-test encoders that rely on superficial local patterns. Prior work augments GNNs with bond/atom chemistry, virtual

2. For our related work, see Maruyama (2026a,b,c); Maruyama and Yasuda (2025); Maruyama (2025a,b,c).

nodes, or domain-inspired descriptors, while data augmentation typically targets permutation and minor attribute noise (Hu et al., 2020; Wu et al., 2018; Duvenaud et al., 2015; Gilmer et al., 2017). Our focus is complementary: we enforce sensitivity to global cycle-space structure (via β_1 and 2-core statistics) and additivity, which standard objectives do not guarantee, without increasing encoder capacity or changing its architecture.

1.1.6. Relation to Prior Work and Novelty

Compared to equivariant architectures (Cohen and Welling, 2016; Bronstein et al., 2021), HomRL introduces a constraint that is (i) architecture-agnostic, (ii) additive and deformation-stable by construction, and (iii) tailored to scaffold-preserving edits common in molecular graphs. Relative to TDA-based pipelines (Carlsson, 2009), our contribution is to *use homological information as supervised targets* for an auxiliary head, thereby shaping the latent representation rather than only enriching inputs. Compared to sheaf-based models (Curry, 2013), we do not require designing a sheaf; instead, we regress a small, computable signature that implicitly captures global structure and yields empirical robustness gains with unchanged model capacity.

2. Graph Homology

This section explains graph homology used throughout, introduces the 2-core and its properties, and formalizes the additive viewpoint we leverage.

We regard a finite, simple, undirected graph $G = (V, E)$ as a 1-dimensional CW complex. Its homology over a field (here $\mathbb{Q}$) is concentrated in degrees 0 and 1:

$$H_0(G; \mathbb{Q}) \cong \mathbb{Q}^{\beta_0(G)}, \qquad H_1(G; \mathbb{Q}) \cong \mathbb{Q}^{\beta_1(G)}.$$

Here $\beta_0(G)$ is the number of connected components and $\beta_1(G)$ is the cycle rank (dimension of the cycle space). The Euler characteristic is $\chi(G) = |V| - |E|$.

Proposition 1 *For any finite graph G, $\beta_1(G) = |E| - |V| + \beta_0(G)$.*

Proof Let F be a spanning forest of G, i.e., the union of one spanning tree per connected component. Then $|E(F)| = |V| - \beta_0(G)$. For each $e \in E \setminus F$, adding e to F creates exactly one *fundamental cycle* C_e (the unique path in F between the endpoints of e plus e). These cycles are linearly independent in the cycle space since each C_e contains the edge e, which no $C_{e'}$ with $e' \neq e$ contains. Hence $\beta_1(G) = |E \setminus F| = |E| - |E(F)| = |E| - (|V| - \beta_0(G))$. ■

The 2-core $G^{(2)}$ is obtained by iteratively removing all vertices of degree < 2 and their incident edges until none remain. Equivalently, $G^{(2)}$ is the unique maximal induced subgraph with minimum degree ≥ 2. A standard queue-based implementation computes $G^{(2)}$ in $O(|V| + |E|)$ time.

Proposition 2 *Let G' be obtained from G by attaching a finite forest $\mathcal{F}$ so that each tree $T \in \mathcal{F}$ meets G in exactly one vertex (either by identifying a vertex of T with a vertex of G or by adding a single edge from a vertex of G to a vertex of T). Then*

$$\beta_1(G') = \beta_1(G), \qquad \chi(G') = \chi(G), \qquad G'^{(2)} \cong G^{(2)}, \qquad \#\mathrm{comp}\big(G'^{(2)}\big) = \#\mathrm{comp}\big(G^{(2)}\big).$$

Moreover, $|V(G'^{(2)})| = |V(G^{(2)})|$. *(The normalized ratio* $|V(G^{(2)})|/|V(G)|$ *may change since* $|V|$ *changes.)*

Proof *Euler and cycle ranks.* For a tree T with $|V(T)| = n$ and $|E(T)| = n - 1$: (i) If a vertex of T is identified with a vertex of G, then $\Delta V = n - 1$ and $\Delta E = n - 1$; hence $|V| - |E|$ is preserved. (ii) If T is kept disjoint and connected to G by a single new edge, then $\Delta V = n$ and $\Delta E = (n - 1) + 1 = n$; again $|V| - |E|$ is preserved. In either case the number of components of G does not change: we either glue at a vertex or add exactly one edge to merge the newly added component with one component of G, never merging two distinct components of G with each other. Therefore $\chi = |V| - |E|$ and $\beta_1 = |E| - |V| + \beta_0$ are both preserved.

2*-core.* The subgraph added is a forest. Any finite forest has a vertex of degree 1; under iterative removal of all vertices of degree < 2, every vertex in each attached tree is eventually deleted (the root may temporarily have degree 2, but once its tree neighbors are removed its degree drops below 2 and it is removed as well). Hence no new vertex from the forest survives in the 2-core, and the induced subgraph on $V(G)$ that remains is exactly $G^{(2)}$. Consequently $G'^{(2)} \cong G^{(2)}$, $|V(G'^{(2)})| = |V(G^{(2)})|$, and $\#\mathrm{comp}(G'^{(2)}) = \#\mathrm{comp}(G^{(2)})$. ■

Given G, the quantities $\beta_0(G)$, $\beta_1(G)$, $\chi(G)$, $G^{(2)}$, $|V(G^{(2)})|$, and $\#\mathrm{comp}(G^{(2)})$ can all be computed in $O(|V| + |E|)$ time via a union–find (for β_0), a spanning forest (for β_1 by Proposition 1), and pruning (for $G^{(2)}$).

Let $\mathcal{G}$ denote isomorphism classes of finite graphs, with disjoint union $\sqcup$ as the additive operation and the empty graph $\emptyset$ as the zero. This makes $(\mathcal{G}, \sqcup, \emptyset)$ a commutative monoid. We use "cut-and-paste" to refer to reasoning that decomposes a graph into a disjoint union of subgraphs and aggregates invariants additively.

Definition 3 (additive measure on graphs) *Let* $\mathcal{G}$ *denote isomorphism classes of finite graphs, with* $\sqcup$ *the disjoint union and* $\emptyset$ *the empty graph. A map* $\mu : \mathcal{G} \to R$ *to a commutative semiring* $(R, +, \cdot)$ *is* additive *if* $\mu(G \sqcup H) = \mu(G) + \mu(H)$ *and* $\mu(\emptyset) = 0$.

For finite graphs, $\mu_{\beta_0}(G) = \beta_0(G), \mu_{\beta_1}(G) = \beta_1(G), \mu_\chi(G) = \chi(G)$ are additive. Since $(G \sqcup H)^{(2)} = G^{(2)} \sqcup H^{(2)}$, the quantities $|V(G^{(2)})|$ and $\#\mathrm{comp}(G^{(2)})$ are also additive under disjoint union. Normalized statistics such as $|V(G^{(2)})|/|V(G)|$ are *not* additive, but are useful as scale-aware summaries. In practice, we package several additive pieces into a vector-valued measure and may optionally include a normalized coordinate for interpretability.

Why does additivity matter? If a representation $z(G)$ is trained so that a linear readout predicts an additive $\mu(G)$, then for disjoint decompositions $G = H \sqcup K$ we expect the readout to respect $\mu(G) = \mu(H) + \mu(K)$. This imposes algebraic structure on $z(\cdot)$ that is stable under common edit operations (Proposition 2) and supports compositional generalization across subgraph assemblies.

3. Homological Representation Learning

This section specifies the *homological signature* used as an auxiliary target, the encoder and training objective, a stability bound linking signature fit to representation invariance, and basic computational considerations. Throughout, $G = (V, E)$ denotes a finite simple graph. $\beta_0(G)$, $\beta_1(G)$, $\chi(G) = |V| - |E|$, and the 2-core $G^{(2)}$ are as in Section 2.

3.1. Homological Signature

We collect homological and core-structure summaries into a small vector

$$s(G) \;=\; \Big(\, \beta_1(G),\; \chi(G),\; \tfrac{|V(G^{(2)})|}{|V(G)|},\; \#\mathrm{comp}\big(G^{(2)}\big) \,\Big) \in \mathbb{R}^4.$$

It has the following properties:

- *Additivity.* For disjoint unions, β_1, χ, $|V(G^{(2)})|$, and $\#\mathrm{comp}(G^{(2)})$ are additive; the normalized ratio $|V(G^{(2)})|/|V(G)|$ is not additive but is scale-aware.
- *Stability to tree edits.* Attaching or pruning forests at existing vertices preserves β_1, χ, $G^{(2)}$, and $\#\mathrm{comp}(G^{(2)})$ (Proposition 2).
- *Computability.* Each entry of $s(G)$ is computable in $O(|V| + |E|)$ time.

In practice, one may use the full $s(G)$ or its additive subvector $(\beta_1, \chi, |V(G^{(2)})|, \#\mathrm{comp}(G^{(2)}))$ depending on the augmentation regime.

3.2. Encoder and Objective

Let $z_\theta(G) \in \mathbb{R}^d$ be the output of a permutation-invariant encoder (e.g., message passing with pooling). We add two linear heads:

$$\hat{y} = W_c z_\theta(G) + b_c, \qquad \widehat{s} = W_m z_\theta(G) + b_m,$$

for classification and homological regression, respectively. Given a label y, the training loss augments the task objective with an ℓ_2 penalty on the signature prediction,

$$\mathcal{L} \;=\; \mathrm{CE}(\hat{y}, y) \;+\; \lambda\, \|\widehat{s} - s(G)\|_2^2 \;+\; \mu \underbrace{\;\mathbb{E}_{a\in\mathcal{A}}\big[\, \|\widehat{s}(a\cdot G) - \widehat{s}(G)\|_2^2 \,\big]\;}_{\text{consistency on chosen augmentations}}, \tag{1}$$

where $\lambda, \mu \geq 0$, and $\mathcal{A}$ is a set of scaffold-preserving augmentations (e.g., node feature noise, branch rewires that do not alter cycles). The consistency term is optional and can be set to zero when not used.

3.3. Representation Invariance Bound

We quantify how accurately fitting the homological head controls the sensitivity of the latent representation to edits that preserve the target signature.

Theorem 4 *Let P be the orthogonal projector onto $(\ker W_m)^\perp$ and let $\sigma^+_{\min}(W_m) > 0$ denote the smallest nonzero singular value of W_m. Assume the head attains uniform error $\|\widehat{s}(H) - s(H)\|_2 \leq \varepsilon$ on a family $\mathcal{F}$. If $G, G' \in \mathcal{F}$ satisfy $s(G) = s(G')$, then*

$$\big\|P\big(z_\theta(G) - z_\theta(G')\big)\big\|_2 \;\leq\; \frac{2\varepsilon}{\sigma^+_{\min}(W_m)}.$$

More generally, with $\delta = \|s(G) - s(G')\|_2$,

$$\big\|P\big(z_\theta(G) - z_\theta(G')\big)\big\|_2 \;\leq\; \frac{2\varepsilon + \delta}{\sigma^+_{\min}(W_m)}.$$

Proof Let $\Delta z = z_\theta(G) - z_\theta(G')$. Affinity of the head gives $W_m \Delta z = \widehat{s}(G) - \widehat{s}(G')$. Triangular inequality yields

$$\|W_m \Delta z\|_2 \le \|\widehat{s}(G) - s(G)\|_2 + \big\|s(G) - s(G')\big\|_2 + \big\|\widehat{s}(G') - s(G')\big\|_2 \le 2\varepsilon + \delta.$$

Write the SVD $W_m = U\Sigma V^\top$ and decompose $V^\top \Delta z = (\xi, \eta)$ with ξ in the span of the columns corresponding to nonzero singular values and η in $\ker W_m$. Then $\big\|\Sigma V^\top \Delta z\big\|_2 \ge \sigma^+_{\min}(W_m)\ \|\xi\|_2$ and $P\Delta z = V_r \xi$, so

$$\|P\Delta z\|_2 = \|\xi\|_2 \le (2\varepsilon + \delta)/\sigma^+_{\min}(W_m).$$

This completes the proof. ■

As a corollary of the above theorem, we obtain:

Corollary 5 *If W_m has full column rank (so $\sigma_{\min}(W_m) > 0$), then $P = I$ and*

$$\big\|z_\theta(G) - z_\theta(G')\big\|_2 \ \le\ \frac{2\varepsilon + \delta}{\sigma_{\min}(W_m)}.$$

For computational considerations, we note the following. (i) *Cost:* Computing $s(G)$ is $O(|V| + |E|)$; the extra head adds $O(d \times 4)$ parameters and negligible runtime. (ii) *Stability:* The regression target is integer- (or count-) valued for three coordinates and a bounded ratio for the fourth; this avoids exploding targets and simplifies tuning λ. (iii) *Implementation:* The method is architecture-agnostic: any encoder that outputs $z_\theta(G)$ can be augmented with the linear head W_m, b_m and trained with (1).

4. Graph Classification Experiment

We validate HomRL on a graph classification task designed for ring-system discrimination in molecular graphs while subjecting models to scaffold-preserving distribution shift. The goal is to isolate the effect of the homological head without changing encoder capacity, training budget, or optimization.

4.1. Task and Data

Task. Binary classification with balanced classes:

- **Class 0 (mono-cyclic):** a single simple cycle of length 5–7 with attached trees (branches) at random vertices; here $\beta_1 = 1$.
- **Class 1 (bi-/fused-cyclic):** two simple cycles (each length 5–7) either disjoint in the same connected component or fused by sharing an edge or a vertex; branches are attached as above; here $\beta_1 \ge 2$.

Node features. Each vertex carries a one-hot atom-type proxy in $\{\mathrm{C}, \mathrm{N}, \mathrm{O}, \mathrm{S}\}$ and a (scaled) degree feature. Small zero-mean noise is added to features during training.

Splits. We use 1,800 training graphs (balanced), 600 clean test graphs, and 600 out-of-distribution (OOD) graphs.

OOD shift. The OOD set increases branch density, perturbs node features more strongly, and preserves cycle rank β_1 and the 2-core structure, stressing robustness to scaffold-preserving perturbations.

Invariants for targets. For each graph G we compute the homological signature

$$s(G) = (\beta_1(G), \chi(G), |V(G^{(2)})|/|V(G)|, \#\mathrm{comp}(G^{(2)})).$$

Computation uses: union–find for β_0, a spanning forest for $\beta_1 = |E| - |V| + \beta_0$, and queue-based pruning for $G^{(2)}$; all are $O(|V| + |E|)$.

4.2. Models and Training Setup

Encoder (shared). A two-layer message passing block with mean pooling:

$$\begin{aligned} h^{(1)} &= \mathrm{ReLU}\big(XW_0 + (AX)U_0 + b_0\big), \\ h^{(2)} &= \mathrm{ReLU}\big(h^{(1)}W_1 + (Ah^{(1)})U_1 + b_1\big), \\ g &= \tfrac{1}{|V|}\sum_{i\in V} h_i^{(2)}, \qquad z_\theta(G) = \mathrm{ReLU}(gW_r + b_r). \end{aligned}$$

Here A is the (binary) adjacency matrix; $W., U., b.$ are learned parameters.

Heads. Both methods use the same encoder. The *baseline* uses a linear classification head $\hat{y} = W_c z_\theta + b_c$. *HomRL* adds a linear homological head $\widehat{s} = W_m z_\theta + b_m$ and trains with the loss in (1) using $\lambda = 0.8$ and $\mu = 0$.

Training. Stochastic gradient descent, batch size 32, 25 epochs, identical data order and initialization across methods.

4.3. Methods and Results

We compare the following two methods:

- **Baseline (no homology):** encoder + classifier trained with cross-entropy only.
- **HomRL (ours):** encoder + classifier + homological head trained with cross-entropy and signature regression; encoder architecture and dimension d are identical to the baseline.

We report accuracy on the training set, clean test set, and OOD set.

Results. As shown in the table below, HomRL improved accuracy by +4.83 points on the clean test set and by +5.50 points under OOD shift while learning to predict the signature with low error. The encoder was unchanged; gains are attributable to the homological regularization rather than capacity increases. The homological regression attained test MSE 0.0905. The learned homological head is well conditioned, with empirical singular values approximately $\sigma_{\min}(W_m) \approx 1.046$ and $\sigma_{\max}(W_m) \approx 1.635$, supporting the stability bound in Theorem 4.

Table 1: Results. Encoder, optimizer, epochs, and batch size are identical across rows.

Model	Train Acc	Test Acc	OOD Acc
Baseline (no homology)	0.6294	0.6517	0.6317
HomRL (ours; λ=0.8)	**0.6922**	**0.7000**	**0.6867**

5. Conclusion

We introduced *Homological Representation Learning* (HomRL), a regularization method that aligns graph-level embeddings with an efficiently computable signature capturing cycle-space and 2-core structure. The approach is architecture-agnostic: any encoder producing a pooled graph embedding can be augmented with a linear *homological head* and trained with the composite objective in (1).

Theoretically, Theorem 4 shows that accurate regression of the signature controls the sensitivity of the latent representation to scaffold-preserving edits, yielding a bound of the form

$$\left\|z_\theta(G) - z_\theta(G')\right\|_2 \;\le\; \frac{2\varepsilon + \delta}{\sigma_{\min}(W_m)}.$$

where ε is the uniform regression error and $\delta = \|s(G) - s(G')\|_2$ measures true signature discrepancy.

Empirically, on a molecular graph task with identical encoders and training budgets, HomRL improves accuracy over a baseline by +4.83 points on the test set and +5.50 points under an out-of-distribution shift that preserves the cycle rank (Table 1). These gains arrive with negligible parameter overhead and are consistent with the invariance mechanism captured by Theorem 4.

We note that regressing an additive, deformation-stable signature $s(G)$ shapes the latent space in ways not enforced by task loss alone, improving robustness to scaffold-preserving perturbations. HomRL can also complement architectural symmetry design and can be combined with permutation- or group-equivariant layers without modification to the training recipe. Note also that the targets are interpretable and computed in $O(|V| + |E|)$ time; tuning requires only a single weight λ (and optionally μ for consistency).

Natural extensions of this work include: (i) augmenting $s(G)$ with finite-dimensional encodings of persistence summaries to capture graded information while retaining stability; (ii) incorporating domain-specific additive descriptors when available; (iii) adding a consistency term when using augmentations that provably preserve the selected signature components; and (iv) extending the recipe beyond graphs, e.g., to images via cubical complexes or to higher-dimensional cell complexes where additional Betti numbers are relevant.

In summary, HomRL provides an effective way to inject homological bias into learned representations. Its theoretical invariance guarantee and empirical gains suggest that efficiently computable additive targets can serve as a reliable lever for improving robustness without increasing model capacity.

References

Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35, 2017.

Vladimir Batagelj and Matjaž Zaveršnik. An $o(m)$ algorithm for cores decomposition of networks. *arXiv preprint cs/0310049*, 2003.

Franziska Bittner. The universal euler characteristic for varieties of characteristic zero. *Compositio Mathematica*, 140(4):1011–1032, 2004.

Michael M. Bronstein, Joan Bruna, Taco S. Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46 (2):255–308, 2009.

Mathieu Carrière, Marco Cuturi, and Steve Oudot. Sliced wasserstein kernel for persistence diagrams. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *PMLR*, pages 664–673, 2017.

Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *PMLR*, pages 2786–2796, 2020.

Taco S. Cohen and Max Welling. Group equivariant convolutional networks. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999. PMLR, 2016.

Justin Curry. Sheaves, cosheaves and applications. *arXiv preprint arXiv:1303.3255*, 2013.

David K. Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Jakob Hansen and Robert Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3(4):315–358, 2019.

Christoph Hofer, Roland Kwitt, and Marc Niethammer. Learning representations of persistence barcodes. *Journal of Machine Learning Research*, 20(126):1–45, 2019.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Yoshihiro Maruyama. Categorical equivariant deep learning: Category-equivariant neural networks and universal approximation theorems. *arXiv.2511.18417*, 2025a.

Yoshihiro Maruyama. Learning with category-equivariant architectures for human activity recognition. *arXiv.2511.01139*, 2025b.

Yoshihiro Maruyama. Learning with category-equivariant representations for human activity recognition. *arXiv.2511.00900*, 2025c.

Yoshihiro Maruyama. Algorithm for interpretable graph features via motivic persistent cohomology. *Springer LNCS*, 2026a.

Yoshihiro Maruyama. Algorithms and complexity results for K-theoretic persistent homology. *Springer LNCS*, 2026b.

Yoshihiro Maruyama. Top-K exterior power persistent homology: Algorithm, structure, and stability. *Springer LNCS*, 2026c.

Yoshihiro Maruyama and Arisa Yasuda. K-theoretic persistent cohomology. In *Proceedings of Machine Learning Research*, 2025.

David W. Matula and Leland L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *Journal of the ACM*, 30(3):417–427, 1983.

Stephen B. Seidman. Network structure and minimum degree. *Social Networks*, 5(3):269–287, 1983.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay S. Pande. Moleculenet: A benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2018.