DIFFERENTIABLE HIGHER-ORDER GRAPH LEARNING VIA SHEAF-INSPIRED LOCAL INCONSISTENCY

Anonymous authors
Paper under double-blind review

ABSTRACT

Topological neural networks (TNNs) extend graph neural networks by modeling higher-order interactions beyond pairwise relations, but their effectiveness critically depends on how higher-order structures are constructed. Existing graph lifting methods typically rely on connectivity or feature similarity, which often fail to capture meaningful relationships, particularly in heterophilous graphs where connected nodes exhibit dissimilar features or labels. To address this issue, we propose a graph lifting framework based on local inconsistency, inspired by cellular sheaf theory, where edge-dependent transformations compare node representations in shared spaces to capture relational patterns beyond similarity. We further introduce a differentiable framework that aggregates edge-level inconsistencies over cycles and employs a soft gating mechanism to construct weighted higher-order structures in an end-to-end manner. The proposed framework is applicable across various topological domains, e.g., simplicial and cellular complexes, and can be integrated with diverse TNN architectures. Extensive experiments demonstrate consistent performance improvements across diverse TNN backbones and a wide range of graph lifting baselines, with particularly strong gains on heterophilous graph datasets. Code will be released upon acceptance.

1 INTRODUCTION

Graph neural networks (GNNs) have emerged as a powerful paradigm for learning from relational data, achieving remarkable success across domains such as social networks, biological systems, and citation graphs (Wu et al., 2020; Zhou et al., 2020; Corso et al., 2024). A key limitation of standard GNNs, however, lies in their reliance on pairwise interactions and local message passing, which restricts them from capturing complex relationships involving multiple entities at once (Gong & Cheng, 2019). Many real-world interactions involve more than two entities and are better represented as group-level relationships among multiple nodes, i.e., higher-order interactions (Battiston et al., 2020). To address this, topological neural networks (TNNs) extend GNNs by operating on higher-order structures, such as hypergraphs, simplicial and cellular complexes, enabling interactions beyond pairwise edges (Feng et al., 2019; Ebli et al., 2020; Hajij et al., 2020; Papillon et al., 2023).

Despite their expressive power, the effectiveness of TNNs critically depends on how the underlying higher-order structure is constructed (Telyatnikov et al., 2025). This process, known as *graph lifting*, transforms an input graph into a higher-order topological domain before learning. Existing lifting methods largely fall into two categories: 1) connectivity-based and 2) feature-based approaches. Connectivity-based methods rely solely on graph topology (e.g., cliques or cycles), implicitly assuming that structural proximity reflects coherent relational signals. Feature-based methods, on the other hand, leverage node features or learned embeddings, typically grouping nodes based on similarity in an embedding space. While these approaches provide practical ways to construct higher-order structures, they fundamentally rely on assumptions that either structural proximity or feature similarity reflects meaningful relationships.

This assumption becomes problematic in heterophilous graphs, where connected nodes often exhibit dissimilar features or labels (Gong et al., 2026; Zhu et al., 2020). For example, survey papers may cite works from diverse areas, and functionally related proteins may interact through complementary roles despite having dissimilar attributes. In such settings, similarity-based criteria fail to capture the underlying relational structure, leading to suboptimal higher-order representations. As a result, constructing meaningful higher-order structures remains a fundamental challenge.

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

Beyond the choice of lifting criterion, another important limitation is that many existing lifting procedures are fixed before training. Higher-order cells are typically constructed by unsupervised rules, such as detecting cliques, cycles, or neighborhoods, and the resulting lifted structure remains unchanged. This is problematic because the choice of lifted domain and lifting procedure can substantially affect TNN performance (Franco et al., 2026), while the constructed higher-order cells are not directly guided by the downstream objective. Consequently, a TNN may be optimized on higher-order structures that are structurally valid but not necessarily useful for the target task.

Recent works address this issue by introducing differentiable graph lifting, where candidate higher-order cells are selected through learnable mechanisms and optimized end-to-end with TNNs (Battiloro et al., 2024; Franco et al., 2026). However, existing differentiable lifting methods still rely primarily on distances in embedding spaces to identify candidate cells, implicitly assuming that meaningful relations are reflected by proximity. While this provides task-adaptive structure construction, the proximity assumption remains problematic when relational patterns are not aligned with feature similarity, as in heterophilous graphs.

In this work, we take a different perspective on graph lifting: instead of constructing higher-order structures based on similarity, we build them based on local relational inconsistency. This allows the model to capture meaningful relational patterns even when neighboring nodes exhibit conflicting or dissimilar signals. Our approach is inspired by *cellular sheaf theory*, which models relationships between nodes through edge-dependent transformations and evaluates consistency in a shared space rather than the raw feature space (Bredon, 2012; Curry, 2014). This way, a learnable notion of local inconsistency captures how node representations disagree after relation-specific transformations.

We further introduce a differentiable higher-order learning framework that aggregates edge-level inconsistencies over cycles to construct higher-order structures. Rather than selecting structures in a discrete manner, we employ a soft gating mechanism that assigns importance weights to candidate cycles, enabling end-to-end optimization. The resulting framework can be applied to a wide range of TNN architectures, including simplicial, cellular, and combinatorial complexes.

Contributions. Our main contributions are summarized as follows: **1)** We propose a novel graph lifting framework based on local relational inconsistency, redefining the criteria for higher-order structure construction beyond connectivity- and similarity-based assumptions. **2)** We develop a sheaf-inspired, differentiable framework that models edge-dependent relations and constructs higher-order structures via soft cycle selection, enabling end-to-end learning. **3)** We demonstrate that the proposed method consistently improves performance across diverse TNN architectures and graph lifting baselines, with significant gains on heterophilous graphs.

2 Related Works

2.1 Topological Neural Networks

Topological neural networks (TNNs) extend GNNs by operating on higher-order structures defined over topological domains, enabling interactions beyond pairwise edges. These models represent data using higher-order cells and encode relationships via incidence relations across dimensions. Information is propagated through incidence matrices across nodes, edges, and higher-order cells with the message-passing paradigm of GNNs. Hypergraph-based models capture group-wise interactions beyond pairwise relations (Wang et al., 2023), while simplicial approaches extend this idea to hierarchical multi-dimensional message passing (Ebli et al., 2020). Formulations based on cellular and combinatorial complexes further relax structural constraints, allowing more flexible higher-order representations (Bodnar et al., 2021a; Hajij et al., 2020).

Despite these advances, the effectiveness of TNNs relies heavily on how the underlying higher-order structure is constructed. Higher-order domains are often predefined or heuristic, which may not align with task-relevant relational patterns. This limitation motivates the need for principled graph lifting methods that can construct meaningful higher-order structures in a data-driven manner.

2.2 Graph Lifting

Graph lifting constructs a higher-order topological domain from an input graph before applying a TNN. Formally, given an input graph $G = (V, E)$, a graph lifting can be viewed as a map $\mathbb{L}$ that transforms G into a higher-order domain $\mathcal{X}$:

$$\mathbb{L} : G \mapsto \mathcal{X} = (\mathcal{X}_0, \mathcal{X}_1, \ldots, \mathcal{X}_K), \tag{1}$$

where $\mathcal{X}_0 = V$, $\mathcal{X}_1 = E$, and $\mathcal{X}_k$ denotes the set of k-dimensional cells. These cells represent higher-order entities beyond pairwise edges, while incidence relations specify how cells of different orders are connected and provide the operators used for message passing in TNNs. Existing graph lifting approaches fall into connectivity- and feature-based structural lifting (Telyatnikov et al., 2025).

Connectivity-based lifting. Connectivity-based methods construct higher-order cells using only the graph topology. Typical examples include clique- and cycle-based lifting. Due to their simplicity, these methods are predominantly used in TNNs (Bodnar et al., 2021a;b; Hajij et al., 2022). However, connectivity-based lifting does not consider node features or task-dependent relational signals. Consequently, they may introduce higher-order cells that are structurally valid but not necessarily informative for downstream learning.

Feature-based lifting. Feature-based methods incorporate node features or learned embeddings when constructing higher-order structures. For example, nodes that are close in the feature or embedding space can be grouped to form candidate higher-order cells (Telyatnikov et al., 2025). Recently, Franco et al. (2026) proposed DiffLift, a differentiable lifting approach that learns higher-order structures based on distances in an embedding space, enabling end-to-end optimization of the lifting process.

Despite their advantages, these methods primarily rely on similarity in the embedding space to construct higher-order cells. This assumption can be limiting in heterophilous graphs, where meaningful relations may occur between nodes with dissimilar features or labels. In contrast, we model local relational inconsistency via sheaf-inspired edge-dependent transformations and aggregate it over cycles, rather than relying on pairwise similarity. This provides a different lifting criterion: a candidate cycle is selected not because its nodes are simply close in the embedding space, but because its edge-wise relational discrepancies indicate that it can form a coherent higher-order structure.

3 Preliminary: Cellular Sheaf on Graphs

Cellular sheaf theory provides a mathematical framework for describing how local signals are related across a graph. A cellular sheaf $\mathcal{F}$ assigns a vector space, called a stalk $\mathcal{F}(u)$, to each node u and a pair of linear maps, called restriction maps ρ, to each edge $(u, v) \in E$:

$$\rho_{u\to(u,v)} : \mathcal{F}(u) \to \mathcal{F}(u, v), \quad \rho_{v\to(u,v)} : \mathcal{F}(v) \to \mathcal{F}(u, v), \tag{2}$$

where $\mathcal{F}(u, v)$ denotes the edge-level stalk. Two nodes are locally consistent if their restricted representations agree, i.e., $\rho_{u\to(u,v)}(\mathbf{s}_u) = \rho_{v\to(u,v)}(\mathbf{s}_v)$, where $\mathbf{s}_u \in \mathcal{F}(u)$ and $\mathbf{s}_v \in \mathcal{F}(v)$ are the signal on node u and v, respectively.

A key aspect of this formulation is that local consistency is evaluated in a shared edge space rather than the original feature space. Node signals are first projected via restriction maps, and consistency is determined after transformation. This is particularly useful in heterophilous graphs, where adjacent nodes may have dissimilar raw features or labels but still exhibit meaningful relational patterns.

In this work, we build on this perspective by using sheaf-inspired restriction maps to define local relational inconsistency. This inconsistency is then used as a criterion for constructing higher-order structures. Figure 1 illustrates this intuition, and the learnable formulation is detailed in Section 4.1.

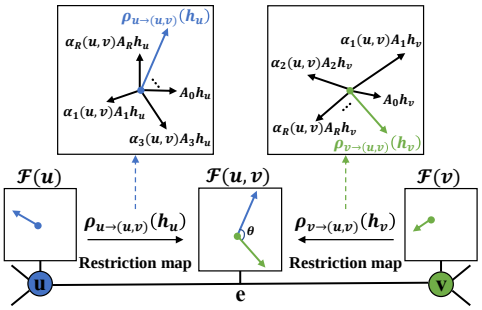

Figure 1: Illustration of sheaf-inspired local inconsistency introduced in Section 4.1. Restriction maps project adjacent node representations h_u, h_v into a shared edge stalk $\mathcal{F}(u, v)$ for comparison.

4 Methods

In this section, we present the proposed framework illustrated in Figure 2. We first define local relational inconsistency through sheaf-inspired edge-dependent transformations. These edge-level inconsistencies are then aggregated over candidate cycles to construct differentiable higher-order structures. Regularization terms are introduced to stabilize cycle selection and relation modeling.

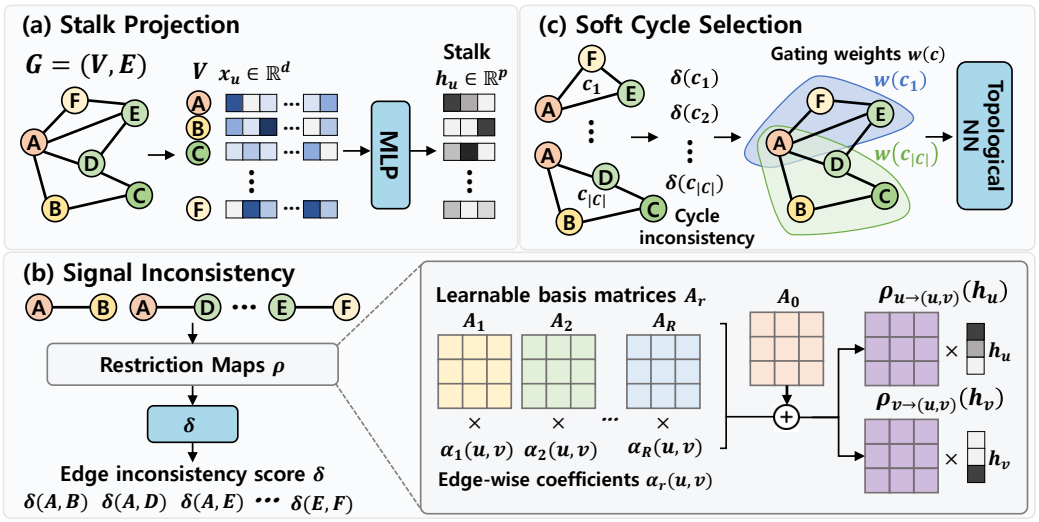

Figure 2: Overview of the proposed framework. (a) Node features are projected into stalk spaces and (b) compared via edge-dependent restriction maps, which model local relational inconsistency through learnable basis transformations. (c) Edge-level inconsistencies are aggregated over cycles to define cycle-level inconsistency scores. A differentiable gating function assigns soft weights to candidate cycles, enabling end-to-end learning of higher-order structures. The resulting weighted complex is then processed by a topological neural network.

4.1 Sheaf-inspired Local Inconsistency

Let $G = (V, E)$ be an undirected graph with node features $x_u \in \mathbb{R}^d$, where d is the feature dimension and $u \in V$. We instantiate the cellular sheaf formulation introduced in Section 3 in a learnable form. In particular, node features are treated as signals on latent stalks, and edge-dependent restriction maps project adjacent node representations into shared edge spaces. Local relational inconsistency is then defined by measuring their disagreement after these transformations.

Stalk Projection. We first map node features x_u into a latent stalk space using a parametric encoder $f_\theta : \mathbb{R}^d \to \mathbb{R}^p$ as

$$h_u = f_\theta(x_u), \quad h_u \in \mathbb{R}^p. \tag{3}$$

To preserve heterophilous signals that may be suppressed by neighborhood aggregation as in conventional GNNs, we adopt node-wise encoders such as multi-layer perceptrons (MLPs). This choice is not restrictive, as any encoder preserving local signal diversity can be utilized.

Sheaf-style Restriction Maps. To model edge-dependent relationships, we introduce learnable restriction maps that project node embeddings into a shared edge space. This construction follows the cellular sheaf principle of mapping signals on incident node stalks into a common edge stalk, where their relational consistency can be evaluated through edge-dependent restriction maps. Instead of assigning independent parameters to each edge, which is computationally burdensome, we propose a basis decomposition that enables flexible yet efficient parameterization. For each edge (u, v), define:

$$\rho_{u\to(u,v)}(h_u) = \left(A_0 + \sum_{r=1}^{R} \alpha_r(u,v) A_r \right) h_u, \tag{4}$$

where $A_0, A_r \in \mathbb{R}^{p\times p}$ are shared learnable basis matrices, $\alpha_r(u, v)$ are edge-dependent coefficients obtained by a neural network conditioned on the pair (h_u, h_v), and R is the number of basis matrices. This formulation allows each edge to induce a distinct transformation while maintaining efficiency with specialized constraints introduced in Section 4.3.

Local Inconsistency via Restricted Agreement. We measure local inconsistency by comparing restricted representations in the shared edge space. Specifically, we define a relation function:

$$\eta(u,v) = \frac{\rho_{u\to(u,v)}(h_u)^\top \rho_{v\to(u,v)}(h_v)}{\|\rho_{u\to(u,v)}(h_u)\| \|\rho_{v\to(u,v)}(h_v)\|}, \tag{5}$$

which measures the agreement between node representations after applying restriction maps. Unlike similarity computed in a shared embedding space, this measure evaluates agreement after edge-dependent transformations, making it relation-dependent rather than globally defined. This can be interpreted as a relaxation of the sheaf consistency condition (Bodnar et al., 2022), where exact equality is replaced by directional alignment in the shared edge space. A high value of $\eta(u, v)$ indicates compatibility under the edge-specific transformation, while a low value indicates a mismatch.

We then define the *local inconsistency score* as:

$$\delta_e(u, v) = 1 - \eta(u, v). \tag{6}$$

This formulation directly reflects the sheaf-theoretic notion of disagreement: $\delta_e(u, v)$ becomes large when the two node signals fail to align after being projected into the same edge space. Importantly, because the restriction maps are edge-dependent, the resulting inconsistency score captures relational discrepancies that are specific to each edge, rather than relying on a global notion of similarity. As a result, $\delta_e(u, v)$ serves as a principled measure of local inconsistency, which is particularly well-suited for heterophilous settings where neighboring nodes may exhibit significantly different signals.

Overall, our approach provides a formulation of the key principles of cellular sheaf theory, i.e., projection into shared spaces and comparison via edge-dependent transformations, in a learnable and scalable manner, while explicitly preserving heterophilous relationships.

4.2 Differentiable Higher-Order Modeling Framework

Given the edge-level inconsistency scores defined in Section 4.1, we construct higher-order structures by aggregating local discrepancies over cycles. Note that we adopt a cycle-based approach, as cycles are naturally defined in graphs and extend to different higher-order domains, including simplicial and cell complexes (Telyatnikov et al., 2025). The choice of this basis structure is flexible, as further discussed in Appendix A.

Cycle Construction and Inconsistency. We first extract a set of candidate cycles $\mathcal{C}$ from the input graph, where each cycle $c \in \mathcal{C}$ is represented as an ordered sequence of nodes forming a cycle. In practice, we obtain $\mathcal{C}$ using a cycle basis with length constraints. To quantify higher-order inconsistency, we aggregate edge-level scores over each cycle. For a cycle c with $|c|$ edges, we define its *inconsistency score* $\delta_c(c)$ as:

$$\delta_c(c) = \frac{1}{|c|} \sum_{(u,v)\in c} \delta_e(u, v). \tag{7}$$

The cycle-level inconsistency $\delta_c(c)$ captures the intuition that a cycle is inconsistent if its constituent edges collectively exhibit disagreement. Compared to edge-level scores, it reflects higher-order structural discrepancies that cannot be detected from pairwise relations alone. Furthermore, $\delta_c(c)$ can be interpreted as a measure of fillability, where cycles with lower $\delta_c(c)$ are more likely to form coherent higher-order structures, while high $\delta_c(c)$ cycles are less suitable for such aggregation.

Soft Cycle Selection. Rather than selecting cycles in a discrete manner, we adopt a differentiable relaxation that assigns a soft importance weight to each cycle via a gating function w:

$$w(c) = \sigma\left(s(\tau - \delta_c(c))\right), \tag{8}$$

where $\tau \in \mathbb{R}$ is a learnable threshold parameter that controls the selection boundary, $s > 0$ is a learnable sharpness parameter, and $\sigma(\cdot)$ denotes the sigmoid function. Cycles with lower inconsistency receive larger weights, while highly inconsistent cycles are suppressed.

Weighted Higher-Order Structure and Feature Construction. We incorporate the selected cycles into the model via a weighted edge-cycle incidence matrix. Let $\mathbf{B}_2 \in \{-1, 0, 1\}^{|E|\times|\mathcal{C}|}$ denote the edge-cycle incidence matrix. Then, the weighted version $\tilde{\mathbf{B}}_2$ can be defined by scaling each column according to the corresponding cycle weight:

$$\tilde{\mathbf{B}}_2 = \mathbf{B}_2 \cdot \mathrm{diag}\big(w(c_1), \ldots, w(c_{|\mathcal{C}|})\big). \tag{9}$$

Thus, each candidate cycle contributes to the lifted structure proportionally to its learned importance.

Proposition 4.1 (Differentiable relaxation of weighted lifting)**.** *With $\delta_c(c)$ and $w(c)$ defined in equation 7 and equation 8, considering $\widetilde{B}_2$ as a function of s and τ, it is differentiable with respect to the cycle inconsistency scores $\{\delta_c(c)\}_{c\in\mathcal{C}}$ and the threshold τ for any finite $s > 0$. Moreover, for all cycles satisfying $\delta_c(c) \neq \tau$, $\lim_{s\to\infty} \widetilde{B}_2 = B_2 \operatorname{diag}\left(\mathbb{I}[\delta_c(c_1) < \tau], \ldots, \mathbb{I}[\delta_c(c_{|\mathcal{C}|}) < \tau]\right)$.*

Proposition 4.1 shows that the proposed weighted incidence matrix is a smooth relaxation of hard threshold-based lifting. For finite s, gradients can flow from the downstream task loss to the cycle weights and inconsistency scores, which enables *end-to-end training* of the lifted higher-order structure and is an essential concept in this work. The proof is provided in Appendix C.

Integration with TNNs. Given node embeddings h_u, edge features can be obtained from the embeddings of the incident nodes. For an edge $e = (u, v)$, we define $h_e = \psi_\theta\left([h_u||h_v||\text{abs}(h_u - h_v)]\right)$, where $||$ is concatenation, $\text{abs}(\cdot)$ is absolute function, and $\psi_\theta(\cdot)$ is a neural network, i.e., an MLP. Using these edge features, cell features can be defined via aggregated cell representation:

$$h_c = \phi_\theta\left(\frac{1}{|c|}\sum_{e\in c} w(c)\cdot h_e\right), \tag{10}$$

where $\phi_\theta(\cdot)$ is a simple MLP.

The resulting node, edge, and cell representations are processed by a topological neural network (TNN), which performs message passing across different dimensions via incidence operators. Because our framework is expressed through weighted incidence matrices, it can be easily integrated with various TNN architectures, including simplicial, cellular, and combinatorial complex models.

Ultimately, the proposed framework enables end-to-end learning of higher-order structures guided by local inconsistency. By softly selecting and weighting cycles, the model adaptively discovers task-relevant topological patterns while remaining fully differentiable.

4.3 Regularizations

The proposed framework jointly learns higher-order structure selection and edge-dependent transformations, but may lead to degenerate or unstable solutions without appropriate regularization.

Gate Sparsity. To prevent trivial solutions where all candidate cycles are selected, we impose a sparsity-inducing regularization on the gating weights:

$$\mathcal{L}_\mathcal{C} = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}} w(c). \tag{11}$$

Since $w(c) \in (0, 1)$ by construction, this term is equivalent to a normalized ℓ_1 penalty on the nonnegative gating weights, encouraging unnecessary cycles to receive near-zero importance.

Basis Orthogonality. To reduce redundancy among the restriction maps, i.e., shared basis matrices $\{A_r\}_{r=1}^R$, we impose an orthogonality constraint:

$$\mathcal{L}_A = \left\|AA^\top - I\right\|_F^2, \tag{12}$$

where A denotes the matrix formed by stacking the vectorized basis matrices. $\mathcal{L}_A$ promotes diversity among basis transformations and stabilizes the learning of edge-dependent mappings.

Relation Entropy. The edge-dependent coefficients $\alpha_r(u, v)$ determine how basis components are combined for each edge. To avoid degenerate solutions where only a small subset of basis components is utilized, we maximize the entropy of the coefficient distribution:

$$\mathcal{L}_\alpha = -\mathbb{E}_{(u,v)}\left[\sum_{r=1}^{R}\alpha_r(u, v)\log\alpha_r(u, v)\right]. \tag{13}$$

This encourages diverse use of the basis components, enabling more expressive relational modeling. The final objective combines the task-specific loss with the proposed regularization terms:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_1\mathcal{L}_\mathcal{C} + \lambda_2\mathcal{L}_A - \lambda_3\mathcal{L}_\alpha, \tag{14}$$

where $\lambda_1, \lambda_2, \lambda_3$ control the strength of each regularization term. Together, these regularizations ensure that the model learns sparse, diverse, and stable higher-order structures while maintaining expressive edge-dependent transformations.

4.4 Computational Complexity

The lifting module adds only linear-time overhead to the TNN backbone: with stalk dimension p and R basis matrices, precomputing $A_r h_u$ for all nodes costs $O(|V|Rp^2)$, after which the edge-level scores in equation 6 cost $O(|E|Rp)$ and cycle scoring and gating in equation 7 and equation 8 cost $O(|\mathcal{C}|L_{\max})$, where $L_{\max}$ is the maximum cycle length. Candidate cycles are extracted once as preprocessing, and the basis decomposition requires only $(R+1)p^2$ parameters for all restriction maps, compared to $O(|E|p^2)$ for edge-independent maps.

5 Experiments

5.1 Datasets and Experimental Setup

Datasets. We evaluate our method on both homophilous and heterophilous node classification benchmarks. For homophilous graphs, we use citation networks including Cora and CiteSeer, where connected nodes tend to share similar labels (Yang et al., 2016). For heterophilous graphs, we use Texas, Cornell, and Actor, where neighboring nodes often belong to different classes, resulting in low homophily and locally conflicting signals (Tang et al., 2009; García-Plaza et al., 2016). These datasets allow us to evaluate whether the proposed lifting captures meaningful relational patterns across both similarity-aligned and conflicting regimes. Due to space constraints, we report the results on Wisconsin (García-Plaza et al., 2016) in Appendix B.3.

Baselines. We compare our method with both standard GNNs and higher-order TNNs. As graph-based baselines, we consider GCN (Kipf & Welling, 2017) and GAT (Veličković et al., 2018), which represent widely used message-passing architectures. To evaluate the effectiveness of our framework in higher-order settings, we adopt several TNN models, including EDGNN (Wang et al., 2023), SCN

Table 1: Quantitative comparison on downstream node classification task. The best scores are highlighted in **Bold**. sheaf denotes the proposed method and OOM refers to out-of-memory.

	Type	Homophilous						Heterophilous								
		Cora			CiteSeer			Texas			Cornell			Actor		
		Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
	GCN	0.849	0.835	0.834	0.700	0.667	0.666	0.553	0.457	0.345	0.553	0.408	0.458	0.341	0.321	0.318
	GAT	0.829	0.816	0.795	0.700	0.669	0.668	0.511	0.338	0.358	0.511	0.310	0.345	0.327	0.316	0.299
EDGNN	khop	0.821	0.804	0.804	0.702	0.675	0.673	0.596	0.439	0.408	0.617	0.480	0.519	0.363	0.348	0.351
EDGNN	cyc.	0.705	0.683	0.678	0.660	0.636	0.634	0.574	0.447	0.388	0.660	0.558	0.573	0.357	0.343	0.342
EDGNN	cliq.	0.712	0.692	0.681	0.671	0.648	0.644	0.553	0.415	0.380	0.702	0.619	**0.605**	0.361	0.347	0.348
EDGNN	kNN	0.679	0.659	0.649	0.628	0.605	0.603	0.596	0.438	0.420	0.617	0.519	0.497	0.369	0.353	0.350
EDGNN	mod.	0.690	0.676	0.662	0.629	0.613	0.600	0.596	0.483	0.408	0.574	0.513	0.498	0.372	0.351	**0.354**
EDGNN	ker.	0.619	0.551	0.526	0.637	0.619	0.610	0.681	0.491	0.498	0.468	0.295	0.276	0.266	0.053	0.200
EDGNN	∂lift	0.810	0.770	0.757	0.734	0.711	0.680	0.757	0.469	0.453	0.649	0.589	0.587	0.356	0.311	0.337
EDGNN	sheaf	**0.875**	**0.865**	**0.863**	**0.756**	**0.738**	**0.705**	**0.786**	**0.561**	**0.536**	**0.716**	**0.624**	0.596	**0.378**	**0.363**	0.352
SCN	khop	0.833	0.830	0.816	0.708	0.679	0.674	0.511	0.444	0.404	0.489	0.345	0.350	OOM	OOM	OOM
SCN	cyc.	0.823	0.816	0.801	0.712	0.687	0.677	0.362	0.257	0.248	0.404	0.216	0.264	0.336	0.315	0.297
SCN	cliq.	0.820	0.812	0.798	0.713	0.688	0.678	0.362	0.257	0.248	0.404	0.216	0.264	0.338	0.315	0.299
SCN	kNN	0.744	0.734	0.705	0.669	0.649	0.635	0.638	0.457	0.456	0.532	0.382	0.404	OOM	OOM	OOM
SCN	mod.	0.843	0.834	0.818	0.707	0.674	0.669	0.511	0.326	0.300	0.489	0.375	0.441	0.332	0.306	0.302
SCN	ker.	0.640	0.624	0.602	0.645	0.624	0.623	0.574	0.424	0.393	0.511	0.414	0.382	OOM	OOM	OOM
SCN	∂lift	0.842	0.836	0.827	0.717	0.693	0.692	0.649	0.380	0.325	0.514	0.491	0.513	0.257	0.271	0.229
SCN	sheaf	**0.858**	**0.846**	**0.847**	**0.734**	**0.704**	**0.694**	**0.744**	**0.468**	**0.482**	**0.668**	**0.537**	**0.531**	**0.376**	**0.332**	**0.347**
CWN	khop	0.854	0.845	0.834	0.711	0.701	0.670	0.511	0.473	0.419	0.489	0.338	0.398	0.348	0.324	0.318
CWN	cyc.	0.838	0.825	0.825	0.720	0.701	0.693	0.617	0.470	0.502	0.574	0.499	0.491	0.359	0.333	0.333
CWN	cliq.	0.851	0.839	0.840	0.724	0.708	0.688	0.447	0.364	0.319	0.447	0.302	0.340	0.359	0.329	0.332
CWN	kNN	0.700	0.684	0.665	0.645	0.621	0.618	0.638	**0.588**	0.529	0.660	0.520	0.551	0.357	0.335	0.338
CWN	mod.	0.830	0.819	0.808	0.725	0.707	0.692	0.553	0.500	0.440	0.489	0.284	0.341	0.341	0.321	0.323
CWN	ker.	0.677	0.662	0.647	0.655	0.637	0.628	0.511	0.533	0.452	0.638	0.501	0.541	0.360	0.339	0.339
CWN	∂lift	0.801	0.789	0.789	0.703	0.680	0.682	0.757	0.422	0.400	0.541	0.497	0.407	0.243	0.236	0.231
CWN	sheaf	**0.875**	**0.866**	**0.861**	**0.753**	**0.720**	**0.708**	**0.808**	0.576	**0.551**	**0.737**	**0.594**	**0.602**	**0.383**	**0.362**	**0.363**
CCXN	khop	0.832	0.824	0.815	0.703	0.672	0.668	0.383	0.314	0.252	0.511	0.481	0.443	0.325	0.296	0.293
CCXN	cyc.	0.846	0.833	0.834	0.711	0.680	0.680	0.447	0.350	0.319	0.404	0.272	0.320	0.321	0.316	0.291
CCXN	cliq.	0.848	0.835	0.836	0.711	0.680	0.680	0.447	0.350	0.319	0.404	0.272	0.320	0.321	0.316	0.291
CCXN	kNN	0.569	0.576	0.507	0.622	0.599	0.594	0.553	0.315	0.387	0.532	0.399	0.424	0.333	0.301	0.303
CCXN	mod.	0.842	0.831	0.826	0.714	0.681	0.680	0.511	0.384	0.377	0.447	0.245	0.284	0.337	0.301	0.300
CCXN	ker.	0.648	0.633	0.620	0.640	0.624	0.605	0.596	0.543	**0.542**	0.617	0.562	0.546	0.343	0.331	0.328
CCXN	∂lift	0.822	0.804	0.809	0.718	0.705	0.670	0.649	0.432	0.408	0.378	0.359	0.420	0.222	0.288	0.219
CCXN	sheaf	**0.873**	**0.870**	**0.860**	**0.753**	**0.708**	**0.704**	**0.794**	**0.564**	0.538	**0.694**	**0.630**	**0.613**	**0.376**	**0.377**	**0.357**

(Ebli et al., 2020), CWN (Bodnar et al., 2021a), and CCXN (Hajij et al., 2020). For higher-order structure construction, we compare with multiple graph lifting baselines, including connectivity-based methods (i.e., k-hop, cycle-, and clique-based lifting) and feature-based approaches (i.e., kNN-, kernel-, and modularity maximization-based lifting), along with differentiable lifting (∂-lifting (Franco et al., 2026)). All baselines are reproduced under our experimental protocol, using the official implementations where available and the TopoBench framework (Telyatnikov et al., 2025) otherwise.

These TNNs are defined on different underlying topological domains, such as simplicial and cellular complexes. By integrating our lifting framework with these diverse architectures, we assess whether the proposed method consistently improves performance across different higher-order representations.

Additional analyses are provided in the appendix: Appendix A studies alternative candidate structures for local-inconsistency lifting, Appendix B reports extended quantitative analyses and ablation results, and Appendix D provides implementation details.

5.2 Experimental results

Classification results. Table 1 presents the quantitative comparison on node classification across both homophilous and heterophilous datasets. On homophilous datasets, our method performs superior to existing approaches, achieving the best results across all metrics, i.e., improving accuracy by $\sim 6.6\%$ compared to the second-best lifting strategy within the same backbone. These results indicate that the proposed framework preserves performance even in settings where local consistency is dominant.

On heterophilous datasets, the advantages of our method become significantly pronounced. Across Texas, Cornell, and Actor, our method consistently outperforms both standard GNNs and higher-order baselines on most metrics, achieving an average improvement of $\approx 10.7\%$ in accuracy over the second-best competing method. These improvements stem from the ability to model relation-specific agreement rather than simple feature similarity, which is often misleading in heterophilous settings.

Compared to the prior differentiable higher-order lifting approach, i.e., DiffLifting, which constructs higher-order structures based on similarity in embedding space, our method explicitly models relational inconsistency via edge-dependent transformations. This allows the model to capture diverse and potentially conflicting local relationships, rather than relying on proximity in a learned feature space. As a result, our approach demonstrates superior performance particularly in heterophilous settings, where similarity-based lifting is inherently limited.

Importantly, our method demonstrates consistent improvements across different topological domains, including hypergraphs (EDGNN), simplicial (SCN), cellular (CWN), and combinatorial (CCXN) complexes, indicating effective generalization of the learned higher-order structures. Hence, these results validate that modeling local inconsistency and incorporating it into a differentiable higher-order framework leads to robust and consistent performance gains, especially in heterophilous settings.

Characterization of lifted 2-cells. To further examine the higher-order structures produced by each lifting method, we characterize the top-20 lifted 2-cells in terms of cycle inconsistency and label composition. Top-k cells are selected via learned weights or importance scores when available; otherwise, we sample k candidate cells and report the average over 50 repetitions. Table 2 reports four metrics on Cora and Cornell, with full metric definitions and results on all datasets provided in Appendix B.1. The average inconsistency $\bar{\delta}_c$ measures how inconsistent the selected 2-cells are under the relational inconsistency function learned by our model. The Low-δ_c ratio measures the fraction of selected 2-cells that satisfy a dataset-specific low-inconsistency criterion. Label purity measures how dominant the majority label is within each 2-cell, while Mixed ratio measures how often a 2-cell contains nodes from multiple classes.

Table 2: Characterization of the lifted 2-cells on Cora and Cornell. We report the mean cycle inconsistency $\bar{\delta}_c$, the fraction of low-inconsistency cells (Low-δ_c ratio), label purity, and the fraction of cells containing multiple classes (Mixed ratio). $\downarrow$ and $\uparrow$ indicate that lower and higher values are better, respectively.

	Cora								Cornell							
Metric	khop	cyc.	cliq.	kNN	mod.	ker.	∂lift	sheaf	khop	cyc.	cliq.	kNN	mod.	ker.	∂lift	sheaf
$\bar{\delta}_c \downarrow$	0.748	0.706	0.575	0.407	0.641	0.336	0.671	0.002	1.123	1.041	0.933	0.705	1.001	1.028	0.715	0.448
Low-δ_c ratio $\uparrow$	0.450	0.468	0.800	0.730	0.450	0.800	0.521	1.000	0.550	0.515	0.750	0.600	0.667	0.400	0.700	1.000
Label purity	0.820	0.933	0.957	0.618	0.961	0.858	0.878	0.983	0.500	0.546	0.583	0.600	0.463	0.683	0.600	0.621
Mixed ratio	0.550	0.186	0.200	0.950	0.150	0.400	0.339	0.050	1.000	0.987	1.000	0.944	1.000	0.900	1.000	0.950

Table 3: Ablation study on regularization. The best scores are highlighted in **Bold**.

	Type	$\mathcal{L}_C$	$\mathcal{L}_A$	$\mathcal{L}_\alpha$	Homophilous Cora Acc.	Cora Prec.	Cora Rec.	Heterophilous Texas Acc.	Texas Prec.	Texas Rec.	Heterophilous Cornell Acc.	Cornell Prec.	Cornell Rec.
EDGNN	None	✗	✗	✗	0.860	0.846	0.849	0.747	0.508	0.508	0.672	0.483	0.519
	w/o $\mathcal{L}_\alpha$	✓	✓		0.870	0.859	0.859	0.780	0.530	0.515	0.682	0.543	0.533
	w/o $\mathcal{L}_A$	✓		✓	0.871	0.862	0.857	0.759	0.537	0.514	**0.727**	0.558	0.564
	w/o $\mathcal{L}_C$		✓	✓	0.873	0.864	0.856	0.770	0.535	0.523	0.711	0.575	0.557
	Full	✓	✓	✓	**0.875**	**0.865**	**0.863**	**0.786**	**0.561**	**0.536**	0.716	**0.624**	**0.596**
CWN	None	✗	✗	✗	0.845	0.839	0.840	0.749	0.454	0.464	0.688	0.486	0.503
	w/o $\mathcal{L}_\alpha$	✓	✓		0.868	0.853	0.855	0.769	0.525	0.502	0.727	0.541	0.579
	w/o $\mathcal{L}_A$	✓		✓	0.865	0.851	0.851	0.765	0.483	0.511	0.728	0.578	0.571
	w/o $\mathcal{L}_C$		✓	✓	0.865	0.857	0.854	0.782	0.528	0.535	0.710	0.544	0.548
	Full	✓	✓	✓	**0.875**	**0.866**	**0.861**	**0.808**	**0.576**	**0.551**	**0.737**	**0.594**	**0.602**
SCN	None	✗	✗	✗	0.847	0.835	0.835	0.677	0.374	0.406	0.602	0.407	0.426
	w/o $\mathcal{L}_\alpha$	✓	✓		0.857	0.839	0.843	0.727	0.436	0.447	0.667	0.456	0.495
	w/o $\mathcal{L}_A$	✓		✓	0.853	0.844	0.841	0.716	0.441	0.453	0.639	0.486	0.499
	w/o $\mathcal{L}_C$		✓	✓	0.852	0.840	0.842	0.715	0.424	0.441	0.639	0.443	0.444
	Full	✓	✓	✓	**0.858**	**0.846**	**0.847**	**0.744**	**0.468**	**0.482**	**0.668**	**0.537**	**0.531**
CCXN	None	✗	✗	✗	0.851	0.842	0.828	0.753	0.500	0.497	0.678	0.481	0.531
	w/o $\mathcal{L}_\alpha$	✓	✓		0.865	0.859	0.850	0.781	0.489	0.507	0.705	0.541	0.552
	w/o $\mathcal{L}_A$	✓		✓	0.866	0.850	0.852	0.771	0.520	0.514	**0.711**	0.573	0.575
	w/o $\mathcal{L}_C$		✓	✓	0.867	0.856	0.854	0.753	0.523	0.513	0.690	0.555	0.565
	Full	✓	✓	✓	**0.873**	**0.870**	**0.860**	**0.794**	**0.564**	**0.538**	0.694	**0.630**	**0.613**

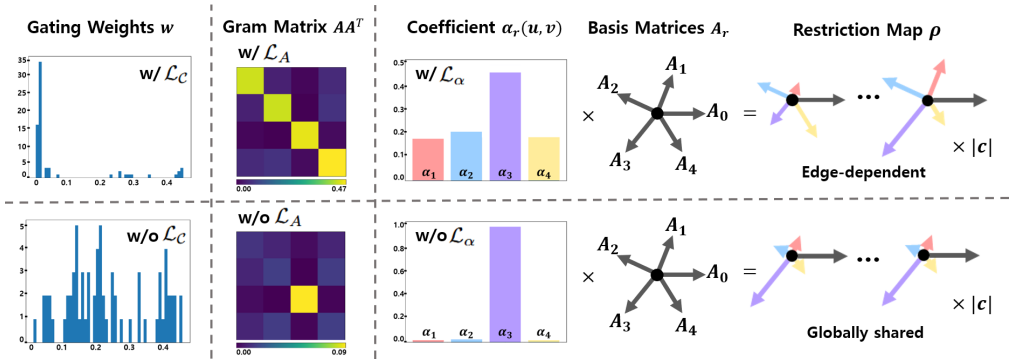

Figure 3: Visualization of the effects of the proposed regularization terms. Top and bottom rows show learned quantities with and without each regularizer. $\mathcal{L}_C$ suppresses unnecessary cycle weights, $\mathcal{L}_A$ promotes diverse, near-orthogonal basis matrices, and $\mathcal{L}_\alpha$ favors diverse relation coefficients for edge-dependent restriction maps.

Across both datasets, our method achieves the lowest $\bar{\delta}_c$ and the highest Low-δ_c ratio, indicating that our method selects more relation-consistent 2-cells than connectivity-, feature-based, and differentiable lifting baselines. On Cora, the selected cells are also highly label-pure, showing that the learned inconsistency criterion aligns with label-homogeneous structures in homophilous graphs. In contrast, on Cornell, our method selects mostly mixed-label 2-cells while still achieving the lowest inconsistency. This suggests that the proposed lifting does not merely favor label-homogeneous structures, but identifies relation-consistent higher-order patterns even under heterophily.

Ablation study on regularization. Table 3 and Figure 3 analyze the effects of the proposed regularization terms, including the cycle gating regularization ($\mathcal{L}_C$), basis regularization ($\mathcal{L}_A$), and entropy regularization on relation coefficients ($\mathcal{L}_\alpha$). Table 3 reports the leave-one-out ablation results. Compared to the unregularized variant (None), the full model consistently improves performance across the reported TNN backbones and datasets, while removing individual terms often leads to degradation. These results provide empirical evidence that the three regularization terms make complementary contributions to higher-order structure learning. Figure 3 further illustrates their intended effects: $\mathcal{L}_C$ promotes compact cycle selection, $\mathcal{L}_A$ reduces redundancy among basis transformations, and $\mathcal{L}_\alpha$ encourages diverse use of relation coefficients, allowing edge-dependent restriction mapping. Together, these regularizers help learn sparse, diverse, and expressive higher-order structures. Full ablation results on more datasets and TNNs are provided in Appendix B.2.

6 Conclusion

We proposed a novel graph lifting framework based on local relational inconsistency, which departs from conventional connectivity- and similarity-based criteria for constructing higher-order structures. By leveraging sheaf-inspired edge-dependent transformations, the method captures relational patterns through disagreement in a shared space, enabling effective modeling even when feature similarity is not informative. We further developed a differentiable framework that aggregates edge-level inconsistencies over cycles with a soft gating mechanism, allowing end-to-end higher-order structure learning. The resulting approach is compatible with a wide range of TNN architectures. Extensive experiments demonstrate consistent performance improvements across multiple backbones and graph lifting baselines, with particularly strong gains on heterophilous graph datasets.

AI USE STATEMENT

Generative AI tools were used only to assist with the writing process, including language refinement, grammar correction, and improving the clarity of the manuscript. They were not used to generate research ideas, conduct experiments, analyze results, or make scientific claims. All AI-assisted text was carefully reviewed and revised by the authors. The authors take full responsibility for the final content of the manuscript.

ETHICS STATEMENT

This work focuses on methodological advances in graph representation learning and does not involve human-subject experiments or new data collection. All experiments are conducted on established graph benchmark datasets. We do not identify specific ethical risks arising directly from the proposed methodology beyond those generally associated with graph-based machine learning systems, such as potential bias in the data or downstream applications. The authors have followed standard research practices in conducting and reporting the experiments.

REPRODUCIBILITY STATEMENT

We provide the implementation and experimental details required to reproduce our results in Appendix D, including the main hyperparameters, candidate structure construction, regularization coefficients, evaluation protocol, and computational environment. Additional ablation studies and extended experimental results are also provided in the appendix. To further support reproducibility, we plan to publicly release the source code upon acceptance.

REFERENCES

Claudio Battiloro et al. From latent graph to latent topology inference: Differentiable cell complex module. In *International Conference on Learning Representations*, 2024.

Federico Battiston et al. Networks beyond pairwise interactions: Structure and dynamics. *Physics reports*, 2020.

Cristian Bodnar et al. Weisfeiler and lehman go cellular: Cw networks. *Advances in Neural Information Processing Systems*, 2021a.

Cristian Bodnar et al. Weisfeiler and lehman go topological: Message passing simplicial networks. In *International Conference on Machine Learning*, 2021b.

Cristian Bodnar et al. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns. *Advances in Neural Information Processing Systems*, 2022.

Glen E Bredon. *Sheaf Theory*. Springer Science & Business Media, 2012.

Gabriele Corso et al. Graph neural networks. *Nature Reviews Methods Primers*, 2024.

Justin Michael Curry. *Sheaves, cosheaves and applications*. University of Pennsylvania, 2014.

Stefania Ebli et al. Simplicial neural networks. In *TDA & Beyond*, 2020.

Yifan Feng et al. Hypergraph neural networks. In *AAAI conference on artificial intelligence*, 2019.

Jorge Luiz Franco et al. Differentiable lifting for topological neural networks. In *International Conference on Learning Representations*, 2026.

Alberto P García-Plaza et al. Using fuzzy logic to leverage html markup for web page representation. *IEEE Transactions on Fuzzy Systems*, 2016.

Cheng-Hua Gong et al. A survey on learning from graphs with heterophily: recent advances and future directions. *Frontiers of Computer Science*, 20(2):2002314, 2026.

Liyu Gong and Qiang Cheng. Exploiting edge features for graph neural networks. In *IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2019.

Mustafa Hajij et al. Cell complex neural networks. In *TDA & Beyond*, 2020.

Mustafa Hajij et al. Topological deep learning: Going beyond graph data. *arXiv preprint arXiv:2206.00606*, 2022.

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

Mathilde Papillon et al. Architectures of topological deep learning: A survey on topological neural networks, april 2023. *arXiv preprint arXiv:2304.10031*, 2023.

Jie Tang et al. Social influence analysis in large-scale networks. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009.

Lev Telyatnikov et al. Topobench: A framework for benchmarking topological deep learning. *Journal of Data-centric Machine Learning Research*, 2025.

Petar Veličković et al. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Peihao Wang et al. Equivariant hypergraph diffusion neural operators. In *International Conference on Learning Representations*, 2023.

Zonghan Wu et al. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 2020.

Zhilin Yang et al. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, 2016.

Jie Zhou et al. Graph neural networks: A review of methods and applications. *AI open*, 2020.

Jiong Zhu et al. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 2020.

A CHOICE OF CANDIDATE STRUCTURES IN LOCAL-INCONSISTENCY LIFTING

Table 4: Ablation on the candidate structure in local-inconsistency lifting.

	Basis	Homophilous						Heterophilous								
		Cora			CiteSeer			Texas			Cornell			Actor		
	struct.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
EDGNN	cyc.	0.875	0.865	0.863	0.756	0.738	0.705	0.786	0.561	0.536	0.716	0.624	0.596	0.378	0.363	0.352
	cliq.	0.869	0.860	0.858	0.751	0.723	0.693	0.770	0.545	0.538	0.720	0.589	0.605	0.376	0.359	0.350
	path	0.866	0.854	0.855	0.751	0.711	0.692	0.794	0.555	0.545	0.733	0.595	0.595	0.371	0.360	0.346
	ego	0.871	0.862	0.855	0.748	0.721	0.693	0.798	0.556	0.549	0.717	0.588	0.566	0.370	0.352	0.353
SCN	cyc.	0.858	0.846	0.847	0.734	0.704	0.694	0.744	0.468	0.482	0.668	0.537	0.531	0.376	0.332	0.347
	cliq.	0.858	0.849	0.846	0.729	0.694	0.699	0.725	0.459	0.474	0.675	0.507	0.528	0.371	0.337	0.338
	path	0.862	0.847	0.853	0.728	0.697	0.690	0.730	0.435	0.456	0.651	0.530	0.495	0.371	0.330	0.334
	ego	0.856	0.839	0.837	0.735	0.694	0.694	0.720	0.431	0.462	0.654	0.508	0.506	0.372	0.312	0.337
CWN	cyc.	0.875	0.866	0.861	0.753	0.720	0.708	0.808	0.576	0.551	0.737	0.594	0.602	0.383	0.362	0.363
	cliq.	0.870	0.858	0.855	0.753	0.721	0.710	0.796	0.594	0.551	0.717	0.538	0.583	0.380	0.360	0.357
	path	0.870	0.857	0.856	0.752	0.718	0.709	0.782	0.474	0.504	0.720	0.561	0.568	0.378	0.362	0.355
	ego	0.870	0.859	0.856	0.753	0.716	0.710	0.815	0.544	0.535	0.724	0.606	0.595	0.374	0.343	0.341
CCXN	cyc.	0.873	0.870	0.860	0.753	0.708	0.704	0.794	0.564	0.538	0.694	0.630	0.613	0.376	0.377	0.357
	cliq.	0.873	0.864	0.860	0.749	0.707	0.692	0.788	0.565	0.524	0.700	0.606	0.628	0.372	0.368	0.357
	path	0.866	0.858	0.852	0.749	0.699	0.694	0.797	0.540	0.516	0.683	0.586	0.614	0.371	0.347	0.347
	ego	0.871	0.858	0.857	0.748	0.715	0.692	0.766	0.540	0.515	0.711	0.597	0.586	0.372	0.357	0.346

This section covers how different basis structures affect local-inconsistency lifting performance. We first describe how each candidate basis structure is constructed for the ablation study. Given a graph $G = (V, E)$, each candidate structure c is represented by a subset of edges $E_c \subseteq E$. The resulting candidate structures are then converted into the edge-cell incidence representation defined in the main text. We consider four types of candidates: cycles, cliques, paths, and ego-networks.

Cycle Candidates. Cycle candidates correspond to closed loop structures in the graph. A cycle candidate is defined as an ordered node sequence $c = (v_1, v_2, \ldots, v_k)$, where consecutive nodes are connected and the last node is connected back to the first node:

$$(v_i, v_{i+1}) \in E \quad \text{for } i = 1, \ldots, k-1, \qquad (v_k, v_1) \in E.$$

The associated edge set is

$$E_c = \{(v_1, v_2), (v_2, v_3), \ldots, (v_{k-1}, v_k), (v_k, v_1)\}.$$

In our implementation, cycle candidates are extracted from a cycle basis of the graph and filtered by a maximum cycle length. These candidates serve as the main closed higher-order structures.

Clique Candidates. Clique candidates represent densely connected local structures. A clique candidate is defined by a node subset $S \subseteq V$ such that every pair of distinct nodes in S is connected:

$$(u, v) \in E \quad \forall u, v \in S,\ u \neq v.$$

The associated edge set is the induced edge set

$$E_c = \{(u, v) \in E \mid u, v \in S\}.$$

In our implementation, maximal cliques are extracted and filtered by a maximum candidate size. Clique candidates serve as an alternative candidate structure that captures densely connected local groups.

Path Candidates. Path candidates are non-closed multi-edge structures. A path candidate is defined as an ordered node sequence $p = (v_1, v_2, \ldots, v_k)$, where consecutive nodes are connected:

$$(v_i, v_{i+1}) \in E \quad \text{for } i = 1, \ldots, k-1.$$

The associated edge set is

$$E_p = \{(v_1, v_2), (v_2, v_3), \ldots, (v_{k-1}, v_k)\}.$$

In our implementation, path candidates are generated from shortest paths between node pairs under a maximum path length constraint. Although paths are not closed, they provide an alternative candidate structure for grouping multiple edges, allowing us to examine how local-inconsistency lifting behaves beyond closed structures.

Ego-Network Candidates. Ego-network candidates represent node-centered local neighborhoods. For each center node $v \in V$, we define its one-hop ego node set as

$$S_v = \{v\} \cup \mathcal{N}(v),$$

where $\mathcal{N}(v)$ denotes the set of one-hop neighbors of v. The associated edge set is the induced edge set on S_v:

$$E_v^{\text{ego}} = \{(u, w) \in E \mid u, w \in S_v\}.$$

Thus, an ego-network candidate includes both center-neighbor edges and edges among the neighbors. If the ego-network is larger than the maximum candidate size, we keep the center node and select a subset of its neighbors. Ego-network candidates serve as an alternative candidate structure that captures node-centered local neighborhoods.

Table 4 compares different candidate structures for local-inconsistency lifting. The results show that the proposed framework achieves competitive performance across diverse candidate structures, including cycles, cliques, paths, and ego-networks. While the best-performing structure varies depending on the dataset and TNN backbone, all candidate types lead to comparable performance in most cases. This indicates that the proposed local-inconsistency criterion is not restricted to cycle-based lifting, but can be applied to different forms of higher-order candidate structures. In the main experiments, we use cycles as a natural default choice because they are naturally defined on graphs and directly correspond to 2-cells in simplicial and cellular complexes, but the framework itself is flexible with respect to the candidate construction rule.

B ADDITIONAL EXPERIMENTS

In this section, we provide additional experimental results omitted from the main manuscript due to space limitations, including quantitative characterization, node classification results, and ablation studies.

B.1 CHARACTERIZATION ON ALL DATASETS

Table 5: Characterization of the lifted 2-cells under a fixed top-k budget.

	Cora								Citeseer							
Metric	khop	cyc.	cliq.	kNN	mod.	ker.	∂lift	sheaf	khop	cyc.	cliq.	kNN	mod.	ker.	∂lift	sheaf
$\bar{\delta}_c(c) \downarrow$	0.748	0.706	0.575	0.407	0.641	0.336	0.671	0.002	0.372	0.321	0.179	0.101	0.180	0.235	0.207	0.004
Low-δ_c ratio $\uparrow$	0.450	0.468	0.800	0.730	0.450	0.800	0.521	1.000	0.450	0.517	0.750	0.860	0.650	0.650	0.716	1.000
Label purity	0.820	0.933	0.957	0.618	0.961	0.858	0.878	0.983	0.891	0.890	0.990	0.721	0.875	0.742	0.892	1.000
Mixed ratio	0.550	0.186	0.200	0.95	0.150	0.400	0.339	0.050	0.450	0.319	0.050	0.850	0.400	1.000	0.294	0.000
	Texas								Wisconsin							
$\bar{\delta}_c(c) \downarrow$	1.383	1.215	1.159	1.244	1.394	1.064	1.225	0.886	1.048	1.093	1.049	1.114	1.142	0.685	1.078	0.458
Low-δ_c ratio $\uparrow$	0.300	0.466	0.650	0.364	0.400	0.550	0.286	1.000	0.450	0.480	0.350	0.400	0.333	0.650	0.200	1.000
Label purity	0.556	0.441	0.429	0.583	0.571	0.567	0.381	0.519	0.625	0.530	0.529	0.662	0.513	0.733	0.467	0.615
Mixed ratio	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.990	1.000	1.000	1.000	0.950	1.000	0.900
	Cornell								Actor							
$\bar{\delta}_c(c) \downarrow$	1.123	1.041	0.933	0.705	1.001	1.028	0.715	0.448	0.765	0.766	0.865	0.346	0.580	0.242	0.708	0.003
Low-δ_c ratio $\uparrow$	0.550	0.515	0.750	0.600	0.667	0.400	0.700	1.000	0.550	0.498	0.450	0.850	0.500	0.850	0.506	1.000
Label purity	0.500	0.546	0.583	0.600	0.463	0.683	0.600	0.621	0.405	0.501	0.450	0.489	0.433	0.517	0.504	0.750
Mixed ratio	1.000	0.987	1.000	0.944	1.000	0.900	1.000	0.950	1.000	0.969	1.000	1.000	1.000	1.000	0.969	0.750

Let $\mathcal{S}_k$ denote the set of top-k lifted 2-cells selected by each method, and let V_c be the set of nodes contained in a 2-cell $c \in \mathcal{S}_k$. For each cell, $\delta_c(c)$ denotes its cycle-level inconsistency score. We compute four statistics as follows.

The average inconsistency evaluates how relation-inconsistent the selected cells are under the learned inconsistency function:

$$\bar{\delta}_c = \frac{1}{|\mathcal{S}_k|} \sum_{c \in \mathcal{S}_k} \delta_c(c).$$

The Low-δ_c ratio measures the fraction of selected cells that satisfy a low-inconsistency criterion:

$$\frac{1}{|\mathcal{S}_k|} \sum_{c \in \mathcal{S}_k} \mathbb{I}[\delta_c(c) \leq \epsilon],$$

where ϵ is a dataset-specific low-inconsistency threshold.

Label purity evaluates how label-homogeneous each selected cell is by measuring the fraction of nodes belonging to the majority label:

$$\frac{1}{|\mathcal{S}_k|} \sum_{c \in \mathcal{S}_k} \max_{\ell} \frac{|\{v \in V_c : y_v = \ell\}|}{|V_c|}.$$

Mixed ratio measures how often selected cells contain nodes from more than one class:

$$\frac{1}{|\mathcal{S}_k|} \sum_{c \in \mathcal{S}_k} \mathbb{I}\left[|\{y_v : v \in V_c\}| > 1\right].$$

Table 5 reports the quantitative analyses of lifted 2-cells on all datasets under a fixed top-k budget. Across datasets, the proposed method consistently selects cells with the lowest average inconsistency and the highest Low-δ_c ratio, indicating that the learned lifting criterion identifies relation-consistent higher-order structures. On homophilous datasets such as Cora and CiteSeer, the selected cells also exhibit high label purity and low mixed-label ratios, suggesting that local relational consistency

aligns well with label-homogeneous structures. In contrast, on heterophilous datasets such as Texas, Cornell, Wisconsin, and Actor, the selected cells often contain mixed labels while maintaining low inconsistency. This supports our claim that the proposed method does not simply favor label-homogeneous cells, but can identify relation-consistent higher-order patterns even when connected nodes have dissimilar labels.

B.2 Ablation results on all datasets

Table 6: Ablation study on regularization. The best scores are highlighted in **Bold**.

	Type	$\mathcal{L}_C$	$\mathcal{L}_A$	$\mathcal{L}_\alpha$	Homophilous						Heterophilous								
					Cora			CiteSeer			Texas			Cornell			Actor		
					Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
EDGNN	None	✗	✗	✗	0.860	0.846	0.849	0.735	0.676	0.684	0.747	0.508	0.508	0.672	0.483	0.519	0.373	0.321	0.341
	w/ $\mathcal{L}_C$	✓			0.864	0.851	0.852	0.735	0.693	0.685	0.759	0.515	0.521	0.701	0.539	0.538	0.378	0.336	0.347
	w/ $\mathcal{L}_A$		✓		0.863	0.855	0.850	0.733	0.693	0.689	0.768	0.514	0.501	0.672	0.524	0.515	0.372	0.331	0.346
	w/ $\mathcal{L}_\alpha$			✓	0.863	0.852	0.846	0.737	0.699	0.688	0.785	0.514	0.522	0.678	0.502	0.531	0.373	0.335	0.349
	w/o $\mathcal{L}_\alpha$	✓	✓		0.870	0.859	0.859	0.744	0.695	0.682	0.780	0.530	0.515	0.682	0.543	0.533	0.377	0.340	0.352
	w/o $\mathcal{L}_A$	✓		✓	0.871	0.862	0.857	**0.757**	0.707	0.696	0.759	0.537	0.514	**0.727**	0.558	0.564	0.377	0.340	0.352
	w/o $\mathcal{L}_C$		✓	✓	0.873	0.864	0.856	0.741	0.698	0.684	0.770	0.535	0.523	0.711	0.575	0.557	0.368	0.354	0.334
	Full	✓	✓	✓	**0.875**	**0.865**	**0.863**	0.756	**0.738**	**0.705**	**0.786**	**0.561**	**0.536**	0.716	**0.624**	**0.596**	**0.378**	**0.363**	**0.352**
SCN	None	✗	✗	✗	0.847	0.835	0.835	0.710	0.672	0.669	0.677	0.374	0.406	0.602	0.407	0.426	0.358	0.304	0.331
	w/ $\mathcal{L}_C$	✓			0.853	0.839	0.844	0.727	0.692	0.685	0.695	0.435	0.436	0.594	0.402	0.448	0.365	0.319	0.335
	w/ $\mathcal{L}_A$		✓		0.852	0.836	0.838	0.729	0.678	0.684	0.687	0.382	0.412	0.601	0.428	0.451	0.366	0.323	0.336
	w/ $\mathcal{L}_\alpha$			✓	0.849	0.831	0.838	0.726	0.675	0.682	0.705	0.425	0.434	0.623	0.413	0.461	0.360	0.310	0.340
	w/o $\mathcal{L}_\alpha$	✓	✓		0.857	0.839	0.843	**0.736**	0.699	0.692	0.727	0.436	0.447	0.667	0.456	0.495	0.369	**0.341**	0.335
	w/o $\mathcal{L}_A$	✓		✓	0.853	0.844	0.841	0.727	0.680	0.680	0.716	0.441	0.453	0.639	0.486	0.499	0.367	0.327	0.332
	w/o $\mathcal{L}_C$		✓	✓	0.852	0.840	0.842	0.728	0.690	0.684	0.715	0.424	0.441	0.639	0.443	0.444	0.375	0.326	0.334
	Full	✓	✓	✓	**0.858**	**0.846**	**0.847**	0.734	**0.704**	**0.694**	**0.744**	**0.468**	**0.482**	**0.668**	**0.537**	**0.531**	**0.376**	0.332	**0.347**
CWN	None	✗	✗	✗	0.845	0.839	0.840	0.717	0.697	0.670	0.749	0.454	0.464	0.688	0.486	0.503	0.376	0.347	0.359
	w/ $\mathcal{L}_C$	✓			0.858	0.844	0.845	0.733	0.686	0.687	0.751	0.486	0.498	0.706	0.497	0.503	0.373	0.338	0.351
	w/ $\mathcal{L}_A$		✓		0.852	0.849	0.842	0.736	0.698	0.694	0.748	0.504	0.474	0.694	0.510	0.510	0.380	0.340	0.362
	w/ $\mathcal{L}_\alpha$			✓	0.857	0.844	0.844	0.736	0.692	0.686	0.754	0.476	0.472	0.700	0.531	0.519	0.384	0.351	0.353
	w/o $\mathcal{L}_\alpha$	✓	✓		0.868	0.853	0.855	0.747	0.704	0.694	0.769	0.525	0.502	0.727	0.541	0.579	0.377	0.358	0.353
	w/o $\mathcal{L}_A$	✓		✓	0.865	0.851	0.851	0.747	0.713	0.689	0.765	0.483	0.511	0.728	0.578	0.571	0.373	0.352	0.355
	w/o $\mathcal{L}_C$		✓	✓	0.865	0.857	0.854	0.749	0.715	0.697	0.782	0.528	0.535	0.710	0.544	0.548	0.375	0.352	0.354
	Full	✓	✓	✓	**0.875**	**0.866**	**0.861**	**0.753**	**0.720**	**0.708**	**0.808**	**0.576**	**0.551**	**0.737**	**0.594**	**0.602**	**0.383**	**0.362**	**0.363**
CCXN	None	✗	✗	✗	0.851	0.842	0.828	0.724	0.683	0.671	0.753	0.500	0.497	0.678	0.481	0.531	0.371	0.350	0.348
	w/ $\mathcal{L}_C$	✓			0.854	0.850	0.831	0.721	0.698	0.686	0.764	0.524	0.519	0.683	0.471	0.513	0.370	0.335	0.348
	w/ $\mathcal{L}_A$		✓		0.858	0.848	0.837	0.720	0.685	0.675	0.765	0.531	0.523	0.688	0.486	0.525	0.370	0.349	0.347
	w/ $\mathcal{L}_\alpha$			✓	0.854	0.835	0.838	0.718	0.686	0.674	0.759	0.518	0.517	0.701	0.504	0.536	0.372	0.356	0.350
	w/o $\mathcal{L}_\alpha$	✓	✓		0.865	0.859	0.850	0.724	0.699	0.685	0.781	0.489	0.507	0.705	0.541	0.552	0.373	0.341	0.348
	w/o $\mathcal{L}_A$	✓		✓	0.866	0.850	0.852	0.742	0.699	0.690	0.771	0.520	0.514	**0.711**	0.573	0.575	0.371	0.338	0.351
	w/o $\mathcal{L}_C$		✓	✓	0.867	0.856	0.854	0.742	0.688	0.690	0.753	0.523	0.513	0.690	0.555	0.565	0.373	0.360	0.351
	Full	✓	✓	✓	**0.873**	**0.870**	**0.860**	**0.753**	**0.708**	**0.704**	**0.794**	**0.564**	**0.538**	0.694	**0.630**	**0.613**	**0.376**	**0.377**	**0.357**

Table 6 reports the full ablation results on all datasets and TNN backbones. Compared to the unregularized variant, the proposed regularization generally improves performance, with the full model achieving the best or highly competitive results in most settings. The single-term and leave-one-out variants suggest that $\mathcal{L}_C$, $\mathcal{L}_A$, and $\mathcal{L}_\alpha$ provide complementary benefits rather than acting as independent replacements for one another. In particular, $\mathcal{L}_C$ stabilizes cycle selection, $\mathcal{L}_A$ reduces redundancy among basis transformations, and $\mathcal{L}_\alpha$ encourages diverse use of basis matrices for edge-dependent restriction maps, as illustrated in Figure 3. These results further support the role of the proposed regularization in learning sparse, diverse, and stable higher-order structures.

B.3 CLASSIFICATION RESULTS ON THE WISCONSIN DATASET

Table 7: Quantitative comparison on downstream node classification task.

Wisconsin	GCN	EDGNN								SCN							
Metric	graph	khop	cyc.	cliq.	kNN	mod.	ker.	∂lift	sheaf	khop	cyc.	cliq.	kNN	mod.	ker.	∂lift	sheaf
Accuracy	0.766	0.813	0.797	0.734	0.813	0.813	0.719	0.745	0.833	0.578	0.453	0.484	0.734	0.578	0.719	0.490	0.793
Precision	0.523	0.724	0.614	0.527	0.741	0.684	0.502	0.691	0.694	0.583	0.331	0.345	0.506	0.376	0.571	0.349	0.597
Recall	0.519	0.647	0.626	0.514	0.724	0.692	0.430	0.706	0.641	0.411	0.300	0.316	0.456	0.359	0.470	0.420	0.588
	GAT	CWN								CCXN							
Accuracy	0.672	0.641	0.766	0.641	0.813	0.625	0.797	0.549	0.864	0.563	0.500	0.500	0.688	0.531	0.797	0.490	0.873
Precision	0.423	0.413	0.518	0.381	0.725	0.390	0.580	0.450	0.701	0.255	0.295	0.295	0.575	0.382	0.668	0.486	0.723
Recall	0.404	0.406	0.501	0.379	0.585	0.384	0.562	0.373	0.695	0.310	0.336	0.336	0.447	0.394	0.580	0.568	0.677

Table 7 reports the node classification results on the Wisconsin dataset, which is additionally evaluated as a heterophilous benchmark. The proposed method consistently achieves the best accuracy across all TNN backbones, improving over both standard GNNs and existing lifting strategies. In particular, the gains are especially clear for SCN, CWN, and CCXN, where the proposed lifting substantially outperforms connectivity-based, feature-based, and differentiable lifting baselines. These results further support the robustness of local-inconsistency-based lifting in heterophilous settings, where feature similarity alone may not provide reliable higher-order structures.

B.4 ABLATION ON THE NUMBER OF BASIS MATRICES IN EQUATION 4

Table 8: Ablation on the number of basis matrices R in Equation 4.

	R	Homophilous						Heterophilous								
		Cora			CiteSeer			Texas			Cornell			Actor		
		Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
EDGNN	4	0.875	0.865	0.863	0.756	0.738	0.705	0.786	0.561	0.536	0.716	0.624	0.596	0.378	0.363	0.352
	8	0.872	0.860	0.857	0.751	0.736	0.700	0.781	0.551	0.531	0.704	0.609	0.582	0.367	0.332	0.340
	16	0.873	0.861	0.859	0.755	0.720	0.697	0.780	0.533	0.512	0.743	0.571	0.587	0.371	0.358	0.345
SCN	4	0.858	0.846	0.847	0.734	0.704	0.694	0.744	0.468	0.482	0.668	0.537	0.531	0.376	0.332	0.347
	8	0.857	0.839	0.842	0.733	0.696	0.688	0.741	0.468	0.469	0.672	0.515	0.507	0.369	0.321	0.336
	16	0.853	0.840	0.841	0.734	0.705	0.693	0.754	0.461	0.485	0.659	0.545	0.545	0.368	0.350	0.334
CWN	4	0.875	0.866	0.861	0.753	0.720	0.708	0.808	0.576	0.551	0.737	0.594	0.602	0.383	0.362	0.363
	8	0.869	0.855	0.855	0.757	0.725	0.706	0.793	0.553	0.556	0.737	0.563	0.594	0.372	0.350	0.340
	16	0.870	0.859	0.859	0.747	0.712	0.695	0.825	0.535	0.543	0.730	0.562	0.614	0.367	0.322	0.334
CCXN	4	0.873	0.870	0.860	0.753	0.708	0.704	0.794	0.564	0.538	0.694	0.630	0.613	0.376	0.377	0.357
	8	0.873	0.870	0.860	0.753	0.708	0.704	0.778	0.550	0.524	0.722	0.570	0.591	0.376	0.349	0.352
	16	0.871	0.858	0.862	0.745	0.712	0.702	0.760	0.553	0.531	0.672	0.573	0.615	0.377	0.375	0.353

Table 8 reports the ablation study on the number of basis matrices R used in the basis decomposition of the restriction maps, introduced in equation 4. Overall, the performance remains stable across different values of R, indicating that the proposed framework is not highly sensitive to this hyperparameter. Increasing R provides more expressive edge-dependent transformations, but does not consistently improve performance across all datasets and backbones. In several cases, a moderate value such as $R = 4$ already achieves the best or highly competitive results, suggesting that a small set of shared basis matrices is sufficient to model diverse relational patterns. These results support the use of a compact basis decomposition, which balances expressiveness and parameter efficiency in the proposed sheaf-inspired restriction maps.

B.5 STD

Table 9: Node classification on homophilous datasets. Mean±std over 10 folds.

		Cora			CiteSeer		
Type		Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
GCN		0.849±0.019	0.835±0.021	0.834±0.023	0.700±0.023	0.667±0.030	0.666±0.024
GAT		0.829±0.018	0.816±0.021	0.795±0.020	0.700±0.021	0.669±0.030	0.668±0.022
EDGNN	khop	0.821±0.024	0.804±0.028	0.804±0.030	0.702±0.017	0.675±0.025	0.673±0.020
	cyc.	0.705±0.025	0.683±0.026	0.678±0.026	0.660±0.020	0.636±0.026	0.634±0.022
	cliq.	0.712±0.018	0.692±0.024	0.681±0.020	0.671±0.022	0.648±0.027	0.644±0.024
	kNN	0.679±0.016	0.659±0.029	0.649±0.018	0.628±0.038	0.605±0.042	0.603±0.037
	mod.	0.690±0.019	0.676±0.028	0.662±0.024	0.629±0.025	0.613±0.031	0.600±0.029
	ker.	0.619±0.023	0.551±0.045	0.526±0.025	0.637±0.034	0.619±0.035	0.610±0.034
	∂lift	0.810±0.018	0.770±0.015	0.757±0.024	0.734±0.024	0.711±0.028	0.680±0.026
	sheaf	0.875±0.023	0.865±0.026	0.863±0.026	0.756±0.021	0.738±0.055	0.705±0.036
SCN	khop	0.833±0.022	0.830±0.026	0.816±0.025	0.708±0.018	0.679±0.028	0.674±0.023
	cyc.	0.823±0.015	0.816±0.018	0.801±0.017	0.712±0.022	0.687±0.031	0.677±0.025
	cliq.	0.820±0.015	0.812±0.020	0.798±0.017	0.713±0.024	0.688±0.034	0.678±0.027
	kNN	0.744±0.020	0.734±0.016	0.705±0.023	0.669±0.023	0.649±0.033	0.635±0.030
	mod.	0.843±0.017	0.834±0.020	0.818±0.018	0.707±0.025	0.674±0.037	0.669±0.030
	ker.	0.640±0.023	0.624±0.032	0.602±0.029	0.645±0.020	0.624±0.030	0.623±0.025
	∂lift	0.842±0.017	0.836±0.016	0.827±0.016	0.717±0.024	0.693±0.033	0.692±0.029
	sheaf	0.858±0.033	0.846±0.032	0.847±0.033	0.734±0.038	0.704±0.038	0.694±0.034
CWN	khop	0.854±0.017	0.845±0.017	0.834±0.020	0.711±0.020	0.701±0.034	0.670±0.025
	cyc.	0.838±0.024	0.825±0.034	0.825±0.024	0.720±0.022	0.701±0.037	0.693±0.024
	cliq.	0.851±0.016	0.839±0.018	0.840±0.022	0.724±0.020	0.708±0.025	0.688±0.024
	kNN	0.700±0.027	0.684±0.028	0.665±0.036	0.645±0.027	0.621±0.033	0.618±0.030
	mod.	0.830±0.024	0.819±0.022	0.808±0.028	0.725±0.018	0.707±0.029	0.692±0.024
	ker.	0.677±0.018	0.662±0.023	0.647±0.022	0.655±0.021	0.637±0.028	0.628±0.021
	∂lift	0.801±0.020	0.789±0.020	0.789±0.028	0.703±0.023	0.680±0.031	0.682±0.027
	sheaf	0.875±0.026	0.866±0.026	0.861±0.027	0.753±0.024	0.720±0.055	0.708±0.025
CCXN	khop	0.832±0.020	0.824±0.024	0.815±0.024	0.703±0.024	0.672±0.044	0.668±0.028
	cyc.	0.846±0.016	0.833±0.021	0.834±0.021	0.711±0.022	0.680±0.041	0.680±0.026
	cliq.	0.848±0.017	0.835±0.022	0.836±0.021	0.711±0.021	0.680±0.038	0.680±0.026
	kNN	0.569±0.028	0.576±0.036	0.507±0.041	0.622±0.025	0.599±0.036	0.594±0.026
	mod.	0.842±0.017	0.831±0.021	0.826±0.019	0.714±0.023	0.681±0.044	0.680±0.026
	ker.	0.648±0.025	0.633±0.037	0.620±0.021	0.640±0.018	0.624±0.032	0.605±0.018
	∂lift	0.822±0.028	0.804±0.025	0.809±0.036	0.718±0.020	0.705±0.025	0.670±0.021
	sheaf	0.873±0.023	0.870±0.023	0.860±0.030	0.753±0.016	0.708±0.052	0.704±0.029

Table 10: Node classification on heterophilous datasets. Mean±std over 10 folds.

		Texas			Cornell			Actor		
Type		Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
GCN		0.553±0.089	0.457±0.133	0.345±0.102	0.553±0.100	0.408±0.125	0.458±0.102	0.341±0.009	0.321±0.013	0.318±0.011
GAT		0.511±0.048	0.338±0.124	0.358±0.101	0.511±0.072	0.310±0.141	0.345±0.121	0.327±0.012	0.316±0.015	0.299±0.015
EDGNN	khop	0.596±0.087	0.439±0.135	0.408±0.112	0.617±0.099	0.480±0.129	0.519±0.155	0.363±0.013	0.348±0.012	0.351±0.013
	cyc.	0.574±0.058	0.447±0.045	0.388±0.050	0.660±0.138	0.558±0.166	0.573±0.183	0.357±0.011	0.343±0.012	0.342±0.011
	cliq.	0.553±0.098	0.415±0.118	0.380±0.125	0.702±0.129	0.619±0.165	0.605±0.179	0.361±0.014	0.347±0.018	0.348±0.016
	kNN	0.596±0.104	0.438±0.100	0.420±0.111	0.617±0.087	0.519±0.144	0.497±0.147	0.369±0.017	0.353±0.020	0.350±0.021
	mod.	0.596±0.108	0.483±0.142	0.408±0.130	0.574±0.128	0.513±0.149	0.498±0.169	0.372±0.019	0.351±0.014	0.354±0.016
	ker.	0.681±0.102	0.491±0.140	0.498±0.117	0.468±0.149	0.295±0.223	0.276±0.145	0.266±0.020	0.053±0.004	0.200±0.000
	∂lift	0.757±0.104	0.469±0.075	0.453±0.048	0.649±0.146	0.589±0.066	0.587±0.037	0.356±0.019	0.311±0.079	0.337±0.011
	sheaf	0.786±0.136	0.561±0.153	0.536±0.137	0.716±0.099	0.624±0.126	0.596±0.124	0.378±0.018	0.363±0.041	0.352±0.016
SCN	khop	0.511±0.099	0.444±0.160	0.404±0.105	0.489±0.102	0.345±0.112	0.350±0.127	OOM	OOM	OOM
	cyc.	0.362±0.076	0.257±0.105	0.248±0.102	0.404±0.076	0.216±0.095	0.264±0.113	0.336±0.014	0.315±0.016	0.297±0.017
	cliq.	0.362±0.095	0.257±0.113	0.248±0.115	0.404±0.079	0.216±0.098	0.264±0.113	0.338±0.017	0.315±0.013	0.299±0.016
	kNN	0.638±0.109	0.457±0.127	0.456±0.084	0.532±0.131	0.382±0.125	0.404±0.158	OOM	OOM	OOM
	mod.	0.511±0.111	0.326±0.145	0.300±0.099	0.489±0.100	0.375±0.151	0.441±0.118	0.332±0.017	0.306±0.014	0.302±0.014
	ker.	0.574±0.068	0.424±0.118	0.393±0.084	0.511±0.064	0.414±0.085	0.382±0.094	OOM	OOM	OOM
	∂lift	0.649±0.113	0.380±0.063	0.325±0.042	0.514±0.110	0.491±0.059	0.513±0.050	0.257±0.014	0.271±0.032	0.229±0.008
	sheaf	0.744±0.102	0.468±0.122	0.482±0.133	0.668±0.142	0.537±0.198	0.531±0.138	0.376±0.022	0.332±0.080	0.347±0.021
CWN	khop	0.511±0.117	0.473±0.171	0.419±0.163	0.489±0.094	0.338±0.125	0.398±0.155	0.348±0.017	0.324±0.019	0.318±0.015
	cyc.	0.617±0.103	0.470±0.116	0.502±0.081	0.574±0.103	0.499±0.097	0.491±0.129	0.359±0.017	0.333±0.018	0.333±0.020
	cliq.	0.447±0.120	0.364±0.106	0.319±0.087	0.447±0.071	0.302±0.113	0.340±0.084	0.359±0.015	0.329±0.017	0.332±0.012
	kNN	0.638±0.085	0.588±0.131	0.529±0.124	0.660±0.122	0.520±0.161	0.551±0.134	0.357±0.020	0.335±0.014	0.338±0.021
	mod.	0.553±0.113	0.500±0.190	0.440±0.148	0.489±0.122	0.284±0.159	0.341±0.160	0.341±0.016	0.321±0.015	0.323±0.016
	ker.	0.511±0.093	0.533±0.166	0.452±0.162	0.638±0.119	0.501±0.121	0.541±0.107	0.360±0.020	0.339±0.016	0.339±0.019
	∂lift	0.757±0.102	0.422±0.074	0.400±0.032	0.541±0.120	0.497±0.072	0.407±0.059	0.243±0.018	0.236±0.066	0.231±0.016
	sheaf	0.808±0.096	0.576±0.127	0.551±0.087	0.737±0.099	0.594±0.117	0.602±0.131	0.383±0.016	0.362±0.031	0.363±0.012
CCXN	khop	0.383±0.113	0.314±0.102	0.252±0.086	0.511±0.090	0.481±0.109	0.443±0.121	0.325±0.019	0.296±0.019	0.293±0.017
	cyc.	0.447±0.092	0.350±0.142	0.319±0.111	0.404±0.131	0.272±0.153	0.320±0.145	0.321±0.015	0.316±0.013	0.291±0.016
	cliq.	0.447±0.092	0.350±0.142	0.319±0.111	0.404±0.131	0.272±0.153	0.320±0.145	0.321±0.015	0.316±0.015	0.291±0.017
	kNN	0.553±0.100	0.315±0.158	0.387±0.157	0.532±0.071	0.399±0.110	0.424±0.104	0.333±0.014	0.301±0.016	0.303±0.014
	mod.	0.511±0.121	0.384±0.099	0.377±0.096	0.447±0.087	0.245±0.112	0.284±0.110	0.337±0.017	0.301±0.017	0.300±0.016
	ker.	0.596±0.076	0.543±0.146	0.542±0.156	0.617±0.084	0.562±0.090	0.546±0.122	0.343±0.017	0.331±0.020	0.328±0.019
	∂lift	0.649±0.125	0.432±0.075	0.408±0.041	0.378±0.119	0.359±0.057	0.420±0.085	0.222±0.012	0.288±0.085	0.219±0.009
	sheaf	0.794±0.081	0.564±0.118	0.538±0.096	0.694±0.094	0.630±0.139	0.613±0.113	0.376±0.016	0.377±0.029	0.357±0.026

C Proof of Proposition 4.1

Proposition C.1 (Differentiable relaxation of weighted lifting)**.** *Let $B_2 \in \{-1, 0, 1\}^{|E|\times|\mathcal{C}|}$ be the edge-cycle incidence matrix, and define $\widetilde{B}_2(s,\tau) = B_2 \operatorname{diag}\big(w(c_1), \dots, w(c_{|\mathcal{C}|})\big)$, where $w(c) = \sigma(s(\tau - \delta_c(c)))$. For any finite $s > 0$, $\widetilde{B}_2(s,\tau)$ is differentiable with respect to the cycle inconsistency scores $\{\delta_c(c)\}_{c\in\mathcal{C}}$ and the threshold τ. Moreover, for all cycles satisfying $\delta_c(c) \neq \tau$, $\lim_{s\to\infty} \widetilde{B}_2(s,\tau) = B_2 \operatorname{diag}\big(\mathbb{I}[\delta_c(c_1) < \tau], \dots, \mathbb{I}[\delta_c(c_{|\mathcal{C}|}) < \tau]\big)$.*

Proof. Let the candidate cycles be indexed as $\mathcal{C} = \{c_1, \dots, c_{|\mathcal{C}|}\}$. For fixed $s > 0$ and τ, the gating weight of each cycle is given by

$$w(c_i) = \sigma\big(s(\tau - \delta_c(c_i))\big).$$

Thus, when viewed as a function of s and τ, the weighted incidence matrix can be written as

$$\widetilde{B}_2(s,\tau) = B_2 \operatorname{diag}\big(w(c_1), \dots, w(c_{|\mathcal{C}|})\big).$$

For any finite $s > 0$, the sigmoid function is smooth, and its argument $s(\tau - \delta_c(c_i))$ is differentiable with respect to $\delta_c(c_i)$ and τ. Therefore, each gating weight $w(c_i)$ is differentiable with respect to $\delta_c(c_i)$ and τ. Since B_2 is fixed and $\widetilde{B}_2(s,\tau)$ is obtained by multiplying B_2 with a diagonal matrix whose entries are differentiable functions, $\widetilde{B}_2(s,\tau)$ is differentiable with respect to $\{\delta_c(c)\}_{c\in\mathcal{C}}$ and τ for any finite $s > 0$.

We next consider the limiting behavior as $s \to \infty$. If $\delta_c(c_i) < \tau$, then $s(\tau - \delta_c(c_i)) \to +\infty$, and hence $w(c_i) \to 1$. If $\delta_c(c_i) > \tau$, then $s(\tau - \delta_c(c_i)) \to -\infty$, and hence $w(c_i) \to 0$. Therefore, for every cycle satisfying $\delta_c(c_i) \neq \tau$,

$$w(c_i) \to \mathbb{I}[\delta_c(c_i) < \tau].$$

Consequently,

$$\lim_{s\to\infty} \widetilde{B}_2(s,\tau) = B_2 \,\mathrm{diag}\left(\mathbb{I}[\delta_c(c_1) < \tau], \ldots, \mathbb{I}[\delta_c(c_{|\mathcal{C}|}) < \tau]\right).$$

This is exactly the incidence matrix obtained by hard thresholding candidate cycles according to $\delta_c(c) < \tau$. □

D IMPLEMENTATION DETAILS

We provide additional details of experiment settings below. The code will be released upon acceptance for better reproducibility. Table 11 summarizes the main hyperparameters used in our experiments. For the sheaf-inspired restriction maps, we use $R = 4$ basis matrices in the main experiments. The edge-dependent coefficients are generated for each edge and used to combine these shared basis matrices, yielding edge-specific restriction maps.

Table 11: Implementation details.

hyperparameter	value
R	4
max_cycle_len	6
k-hop	1
kNN	3
λ_C	1e-3
λ_A	1e-3
λ_α	1e-4

Candidate higher-order structures are constructed from the input graph before differentiable lifting. For cycle-based lifting, we extract candidate cycles with a maximum cycle length of 6. For connectivity-based baselines, we use $k = 1$ for k-hop lifting, while $k = 3$ is used for kNN-based lifting, following Telyatnikov et al. (2025). The proposed differentiable lifting module assigns soft weights to candidate cycles according to their local-inconsistency scores and constructs the corresponding weighted incidence matrix.

The final training objective combines the task loss with the proposed regularization terms. We set the cycle sparsity coefficient to $\lambda_C = 10^{-3}$, the basis regularization coefficient to $\lambda_A = 10^{-3}$, and the relation entropy coefficient to $\lambda_\alpha = 10^{-4}$. These regularization terms control the sparsity of selected cycles, the diversity of basis matrices, and the utilization of relation coefficients, respectively. All experimental results are obtained via 10-fold cross validation for fair comparison, using a single GeForce RTX 3090 with 24GB of GPU memory.