Adaptive Gated Simplicial Propagation for Node Classification in Multimodal Graphs

Linlin Ye[1] **Fangfang Li**[1,2*] **Huihui Zhang**[1] **Lincheng Jiang**[3] **Wei Wu**[1*]

[1]School of Computer Science and Engineering, Central South University
[2]Xiangjiang Laboratory
[3]College of Advanced Interdisciplinary Studies, National University of Defense Technology
{yelinlin, lifangfang, zhanghuihui}@csu.edu.cn
linchengjiang@nudt.edu.cn
william.third.wu@gmail.com

Abstract

Node classification in multimodal graphs plays an important role in many real-world applications, where nodes are described by multiple modalities such as text and images, and the graph structure captures relations among entities. However, most existing Graph Neural Networks (GNNs) still focus on pairwise connections, overlooking higher-order relational patterns. Recent studies have explored simplicial complexes to capture higher-order interactions and integrated them into GNN frameworks. Motivated by this line of work, we propose Adaptive Gated Simplicial Propagation (AGSP), an end-to-end framework for node classification in multimodal graphs. Specifically, AGSP first introduces a simplicial propagation layer to capture higher-order relations beyond pairwise connections; then it applies an adaptive gating mechanism to balance multi-order topological relations, enhancing the discriminative ability of node classification. Extensive experiments validate that AGSP generally outperforms the state-of-the-art baselines, highlighting the effectiveness of combining multimodal information with higher-order graph topology through adaptive gated fusion.

1 Introduction

Multimodal graphs, e.g., social networks, product co-purchase networks, multimedia knowledge graphs, etc., are widely used to model real-world scenarios in which entities are associated with multiple data modalities. Node classification in multimodal graphs supports a wide range of applications, including product categorization in e-commerce networks [1], entity understanding in multimedia knowledge graphs [2], and content analysis on social platforms [3]. In these scenarios, nodes carry textual and visual attributes, while the graph structure may encode higher-order topological patterns that cannot be fully captured by pairwise dependencies, because they only model node-to-node interactions [4–8]. Therefore, effective node classification in multimodal graphs requires modeling multimodal attributes and higher-order structures beyond pairwise relations.

Most existing multimodal GNNs [9–11] are restricted to pairwise message passing over ordinary graphs. While these designs are effective for local neighborhood aggregation, they are limited to pairwise interactions, which prevents them from capturing latent higher-order dependencies. This limitation becomes especially critical for node classification, where information relevant to class labels may emerge from local interactions beyond pairwise links. For example, in multimodal product

*Fangfang Li and Wei Wu are corresponding authors of this work.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

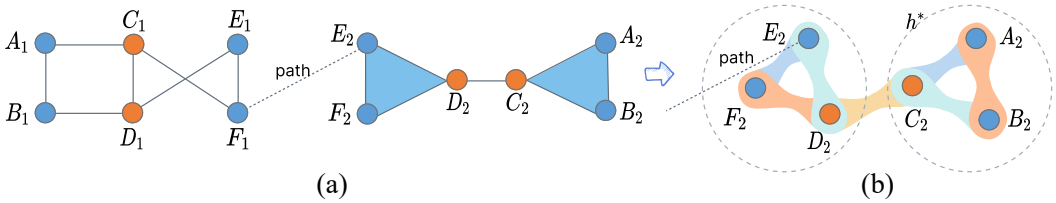


Figure 1: Nodes C_1 and C_2 share isomorphic pairwise interaction patterns but differ in higher-order contexts. Simplicial complexes in (a) distinguish C_1 from C_2 by modeling the 2-simplex $[A_2, B_2, C_2]$, which represents a latent joint intent among the three items beyond pairwise interactions. In contrast, hypergraphs in (b) require such a joint intent to be explicitly given as a hyperedge; the dashed circle labeled h^* denotes a latent candidate hyperedge (A_2, B_2, C_2), which is not predefined because this joint intent is not observed as an explicit group relation in the original graph.

graphs, higher-order relations may arise from groups of products that are jointly co-viewed [12], co-purchased [13], or associated with the same shopping intent [14, 15], rather than from isolated product-to-product interactions. Such higher-order, local patterns often reveal label-relevant semantic context that cannot be captured by pairwise connections [16]. Thus, GNNs relying solely on pairwise message passing cannot explicitly exploit higher-order relations for node classification.

To address these problems, higher-order relational modeling beyond ordinary graphs is needed. Although hypergraphs can connect multiple nodes through hyperedges [17–21], they are mainly suited for modeling explicitly specified group relations. However, in many multimodal graphs, useful higher-order relations, such as latent joint intents among multiple items, may remain implicit in graph topology rather than being predefined as hyperedges [22, 23]. In such cases, simplicial complexes [24–30] can better capture higher-order relations. As shown in Figure 1(a), nodes C_1 and C_2 are indistinguishable from a pairwise perspective due to their isomorphic node-to-node interaction patterns. However, the nodes A_2, B_2, and C_2 form a 2-simplex $[A_2, B_2, C_2]$, which represents a latent joint intent among the three items, such as a potential co-purchase intention, beyond pairwise interactions. This higher-order context helps distinguish C_2 from C_1. In contrast, the hypergraph in Figure 1(b) relies on predefined hyperedges to model group-wise relations. Since the latent hyperedge $h^* = (A_2, B_2, C_2)$ for the potential co-purchase intention is unknown in the original graph, this implicit higher-order relation remains uncaptured, leaving C_1 and C_2 indistinguishable.

Nevertheless, applying simplicial complexes to multimodal graph learning remains largely underexplored. Although simplicial structures can naturally characterize higher-order relations induced by graph topology, how to effectively integrate them with multimodal propagation for node classification is still an open and important problem. To this end, we propose Adaptive Gated Simplicial Propagation (AGSP), an end-to-end framework for node classification in multimodal graphs. AGSP introduces a simplicial propagation layer to capture higher-order relations beyond pairwise connections; further, it designs an adaptive simplicial dependency mechanism to dynamically calibrate the contributions of multi-order topological relations according to the local topological context of each node. In this way, AGSP selectively emphasizes the most informative dependencies for each node, thereby enhancing the discriminative capacity of node classification. The main contributions of our work are summarized as follows.

- To the best of our knowledge, we are the first to introduce simplicial complex modeling into multimodal graph learning. This framework captures higher-order relations that are overlooked by conventional pairwise message passing.
- We propose Adaptive Gated Simplicial Propagation (AGSP), which applies an adaptive gating mechanism to balance multi-order topological relations, enhancing the discriminative ability of node classification.
- We conduct extensive experiments on four real-world datasets. The results show that AGSP outperforms the state-of-the-art baselines, highlighting the effectiveness of combining multimodal information with higher-order graph topology through adaptive gated fusion. The code is publicly available on GitHub[2].

[2] `https://github.com/AIandBD/AGSP`

2 Related Works

Multimodal graph learning. Multimodal graph learning focuses on learning from the integration of graph structure and multimodal node attributes, such as text and images [1, 3, 31, 32]. Early studies [33, 34] typically combine modality-specific encoders with GNNs for recommendation and classification. More recent methods have extended this paradigm to node classification [35], unified embedding learning [1], and cold-start scenarios [36]. Despite their effectiveness, most existing multimodal graph models are still defined on ordinary graphs. Therefore, they mainly rely on pairwise message passing over nodes and edges. As a result, they are effective at modeling local pairwise dependencies but struggle to capture higher-order structural dependencies in multimodal graphs.

Higher-order graph learning. Higher-order graph learning aims to model relational patterns beyond pairwise edges [37] by introducing structures such as hypergraphs [38] and simplicial complexes [24]. Hypergraph learning models group-wise interactions by allowing each hyperedge to connect multiple nodes. Such models have been applied in both general and multimodal settings [4, 5]. However, hypergraphs are mainly suited for explicitly specified higher-order relations and are less effective for capturing latent higher-order dependencies implicit in the graph structure [37, 38]. In contrast, simplicial complexes provide a downward-closed higher-order structure, making them better suited for modeling interactions of different orders within a unified topological framework [24, 37]. Recent simplicial neural networks [24, 25, 39] have demonstrated strong expressive and empirical performance. Nevertheless, existing simplicial methods have mostly been developed for general higher-order learning or non-multimodal settings rather than for multimodal graphs. In particular, building meaningful simplex structures from multimodal node features and leveraging structural dependencies of various orders in multimodal graphs remains underexplored.

3 Preliminaries

In this section, we introduce the required preliminary knowledge.

3.1 Node Classification in Multimodal Graphs

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a multimodal graph, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is the node set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set, with N representing the total number of nodes. $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N] \in \{0, 1\}^{N \times C}$ denotes the label matrix of $\mathcal{G}$, and $\mathbf{y}_i$ is the label vector of node v_i, where C is the number of classes. Each node $v_i \in \mathcal{V}$ is associated with M modalities, denoted by $\{\mathbf{x}_i^{(m)}\}_{m=1}^{M}$, where $\mathbf{x}_i^{(m)} \in \mathbb{R}^{d_m}$ is the d_m-dimensional feature vector of the m-th modality. Let $\mathbf{X}^{(m)} \in \mathbb{R}^{N \times d_m}$ denote the feature matrix of the m-th modality. In this work, nodes are associated with textual and visual attributes, though the formulation applies more generally to multimodal settings. Given the multimodal graph $\mathcal{G}$ and the node features $\{\mathbf{X}^{(m)}\}_{m=1}^{M}$, node classification aims to learn an end-to-end predictor:

$$f_\theta : \left(\mathcal{G}, \{\mathbf{X}^{(m)}\}_{m=1}^{M}\right) \mapsto \hat{\mathbf{Y}},$$

where $\hat{\mathbf{Y}} = [\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_N]^\top \in \mathbb{R}^{N \times C}$ is the predicted label distribution matrix over all nodes. The parameters θ are learned on labeled nodes $\mathcal{V}_{\text{train}} \subseteq \mathcal{V}$ by minimizing the cross-entropy loss:

$$\mathcal{L} = -\sum_{v_i \in \mathcal{V}_{\text{train}}} \mathbf{y}_i^\top \log \hat{\mathbf{y}}_i,$$

where $\hat{\mathbf{y}}_i$ denotes the predicted probability vector for node v_i, and $\top$ denotes the vector transpose.

3.2 Simplicial Complex

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set, a k-simplex is an unordered set of $k+1$ distinct vertices, denoted as $\sigma^k = [v_{i_0}, \ldots, v_{i_k}] \subseteq \mathcal{V}$ [3], where $i_0, i_1, \ldots, i_k$ are distinct

[3]For computational purposes, a k-simplex is an unordered set of $k+1$ vertices; we can encode it as a binary vector $\boldsymbol{\chi}_{\sigma^k} \in \{0,1\}^N$ under a fixed node ordering, where $(\boldsymbol{\chi}_{\sigma^k})_a = \begin{cases} 1, & v_a \in \sigma^k, \\ 0, & v_a \notin \sigma^k. \end{cases}$

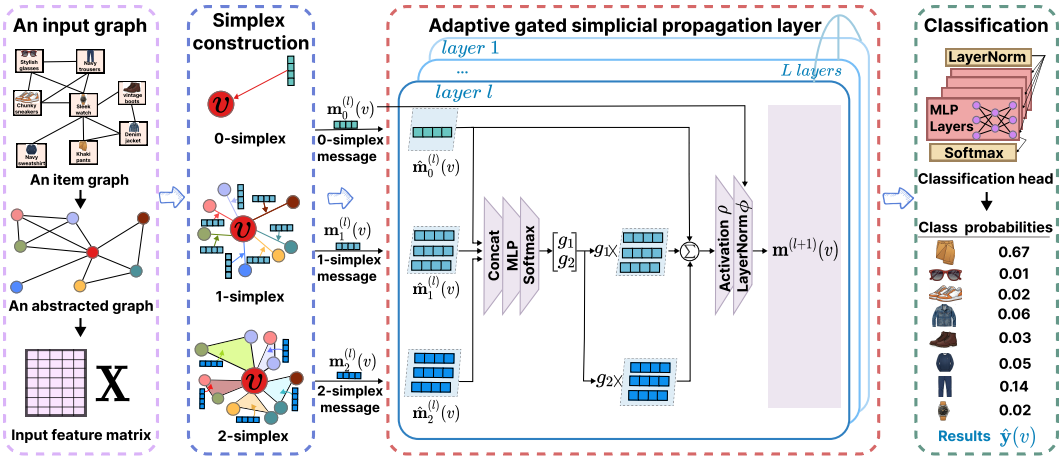


Figure 2: Overview of the proposed AGSP framework.

indices and k is its dimension. In particular, 0-simplex, 1-simplex, and 2-simplex correspond to nodes, edges, and filled triangles, respectively. A face of a simplex is any non-empty subset of its vertices. A simplicial complex $\mathcal{K}$ is a collection of simplices satisfying the downward-closure property, which means that if a simplex belongs to $\mathcal{K}$, then all of its non-empty faces also belong to $\mathcal{K}$. In this work, we define $\mathcal{K} = (\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_K)$ as the simplicial complex constructed from the input graph, where $\mathcal{S}_k$ denotes the set of k-simplices for $k \in \{0, 1, \ldots, K\}$, and K is the maximum simplex dimension considered. We construct $\mathcal{K}$ by clique-complex lifting, which maps each $(k+1)$-clique in $\mathcal{G}$ to a k-simplex in $\mathcal{K}$. The detailed formalization is provided in Appendix A.

4 Adaptive Gated Simplicial Propagation

In this section, we propose a novel end-to-end framework called AGSP for multimodal graphs. AGSP employs simplicial complexes to capture higher-order relations beyond pairwise connections. An adaptive gating mechanism is then used to balance multi-order topological relations for more discriminative node classification. Figure 2 shows the overview of the proposed AGSP algorithm.

4.1 The AGSP Algorithm

Multi-order simplicial feature construction. AGSP computes node-wise features of different simplex orders from multimodal attributes and the lifted simplicial complex $\mathcal{K} = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2)$. For each $v_i \in \mathcal{S}_0$, the 0-simplex feature is initialized by concatenating its textual and visual features:

$$\mathbf{m}_{0,i}^{(0)} = [\mathbf{x}_i^{(1)} \| \mathbf{x}_i^{(2)}], \qquad \forall v_i \in \mathcal{S}_0. \tag{1}$$

This step integrates textual and visual attributes into a unified node-level feature.

Based on the current 0-simplex features, AGSP constructs 1-simplex features for each node by aggregating from its incident edges, which correspond to 1-simplices in $\mathcal{K}$. For notational convenience, we denote the pairwise interaction induced by a 1-simplex $[v_i, v_j]$ as $\mathbf{m}_{1,ij}^{(l)} = \mathbf{m}_{0,i}^{(l)} \odot \mathbf{m}_{0,j}^{(l)}$. When a node has no incident k-simplices for $k \in \{1, 2\}$, the corresponding aggregated message is set to $\mathbf{m}_{k,i}^{(l)} = \mathbf{0}$, and the attention normalization over the empty incident set is omitted. Then, AGSP computes the 1-simplex feature of v_i via attention aggregation over its incident 1-simplices:

$$\mathbf{m}_{1,i}^{(l)} = \sum_{[v_i, v_j] \in \mathcal{S}_1} \alpha_{ij}^{(l)} \mathbf{U}_1 \mathbf{m}_{1,ij}^{(l)}, \tag{2}$$

where $\alpha_{ij}^{(l)}$ is the attention weight of the incident 1-simplex $[v_i, v_j]$ for node v_i, computed by

$$\alpha_{ij}^{(l)} = \frac{\exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}_1^\top [\mathbf{U}_0 \mathbf{m}_{0,i}^{(l)} \| \mathbf{U}_1 \mathbf{m}_{1,ij}^{(l)}]\right)\right)}{\sum_{[v_i, v_r] \in \mathcal{S}_1} \exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}_1^\top [\mathbf{U}_0 \mathbf{m}_{0,i}^{(l)} \| \mathbf{U}_1 \mathbf{m}_{1,ir}^{(l)}]\right)\right)}, \tag{3}$$

Algorithm 1 The AGSP algorithm

Input: Multimodal graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, textual features $\mathbf{X}^{(1)}$, visual features $\mathbf{X}^{(2)}$, propagation depth L
Output: Predicted probability vector $\hat{\mathbf{y}}_i$ for $v_i \in \mathcal{V}$

Construct the simplicial complex $\mathcal{K} = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2)$ from $\mathcal{G}$.
Initialize learnable parameters $\mathbf{U}_0, \mathbf{U}_1, \mathbf{U}_2, \mathbf{a}_1, \mathbf{a}_2, \mathbf{W}_0, \mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_g, \mathbf{b}_g$.
Initialize 0-simplex node features $\mathbf{m}_{0,i}^{(0)}$ via Eq. (1).
for $l = 1, \ldots, L$ **do**
  for each node $v_i \in \mathcal{V}$ **do**
    Compute attention-based 1-simplex node feature $\mathbf{m}_{1,i}^{(l)}$ via Eqs. (2) and (3).
    Compute attention-based 2-simplex node feature $\mathbf{m}_{2,i}^{(l)}$ via Eqs. (4) and (5).
    Project 0-simplex, 1-simplex, and 2-simplex node features via Eq. (6).
    Compute adaptive gate coefficients $g_{1,i}^{(l)}, g_{2,i}^{(l)}$ via Eq. (7).
    Compute gated simplicial feature $\hat{\mathbf{m}}_i^{(l)}$ via Eq. (8).
    Update 0-simplex node feature $\mathbf{m}_{0,i}^{(l+1)} = \phi\left(\mathbf{m}_{0,i}^{(l)}, \hat{\mathbf{m}}_i^{(l)}\right)$.
  end for
end for
Predicted probability vector $\hat{\mathbf{y}}_i = \mathrm{Softmax}\left(\mathrm{MLP}\left(\mathbf{m}_{0,i}^{(L)}\right)\right)$.
return $\hat{\mathbf{y}}_i$

where $\mathbf{U}_0$, $\mathbf{U}_1$, and $\mathbf{a}_1$ are learnable parameters for 1-simplex attention.

For each node v_i, AGSP further computes the 2-simplex node feature by attentively aggregating higher-order interactions from its incident 2-simplices. For each 2-simplex $[v_i, v_j, v_k] \in \mathcal{S}_2$, its higher-order interaction is computed by $\mathbf{m}_{2,ijk}^{(l)} = \mathrm{MaxPool}\left(\mathbf{m}_{1,ij}^{(l)}, \mathbf{m}_{1,ik}^{(l)}, \mathbf{m}_{1,jk}^{(l)}\right)$. Then, the 2-simplex node feature of v_i is obtained by

$$\mathbf{m}_{2,i}^{(l)} = \sum_{[v_i, v_j, v_k] \in \mathcal{S}_2} \beta_{ijk}^{(l)} \mathbf{U}_2 \mathbf{m}_{2,ijk}^{(l)}, \tag{4}$$

where $\beta_{ijk}^{(l)}$ is the attention weight of the incident 2-simplex $[v_i, v_j, v_k]$ for node v_i, computed by

$$\beta_{ijk}^{(l)} = \frac{\exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}_2^\top [\mathbf{U}_0 \mathbf{m}_{0,i}^{(l)} \| \mathbf{U}_2 \mathbf{m}_{2,ijk}^{(l)}]\right)\right)}{\sum_{[v_i, v_p, v_q] \in \mathcal{S}_2} \exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}_2^\top [\mathbf{U}_0 \mathbf{m}_{0,i}^{(l)} \| \mathbf{U}_2 \mathbf{m}_{2,ipq}^{(l)}]\right)\right)}, \tag{5}$$

where $\mathbf{U}_0$, $\mathbf{U}_2$, and $\mathbf{a}_2$ are learnable parameters for 2-simplex attention.

Adaptive gated simplicial propagation. After constructing simplex features, AGSP performs L layers of adaptive gated propagation. At layer l, the multi-order simplicial messages associated with node v_i are first projected into a shared hidden space:

$$\hat{\mathbf{m}}_{0,i}^{(l)} = \mathbf{W}_0 \mathbf{m}_{0,i}^{(l)}, \quad \hat{\mathbf{m}}_{1,i}^{(l)} = \mathbf{W}_1 \mathbf{m}_{1,i}^{(l)}, \quad \hat{\mathbf{m}}_{2,i}^{(l)} = \mathbf{W}_2 \mathbf{m}_{2,i}^{(l)}, \tag{6}$$

where $\hat{\mathbf{m}}_{k,i}^{(l)}$ denotes the projected k-simplex feature of node v_i for $k \in \{0, 1, 2\}$.

Since the contributions of different simplex orders may vary across nodes, AGSP uses an adaptive gate to assign node-specific fusion weights. The gate coefficients are computed as

$$[g_{1,i}^{(l)}, g_{2,i}^{(l)}] = \mathrm{Softmax}\left(\mathbf{W}_g [\hat{\mathbf{m}}_{0,i}^{(l)} \| \hat{\mathbf{m}}_{1,i}^{(l)} \| \hat{\mathbf{m}}_{2,i}^{(l)}] + \mathbf{b}_g\right), \tag{7}$$

where $g_{1,i}^{(l)}$ and $g_{2,i}^{(l)}$ denote the node-specific weights assigned to the 1-simplex and 2-simplex features, respectively. The gated simplicial feature of node v_i is computed as

$$\hat{\mathbf{m}}_i^{(l)} = \rho\left(\hat{\mathbf{m}}_{0,i}^{(l)} + g_{1,i}^{(l)} \hat{\mathbf{m}}_{1,i}^{(l)} + g_{2,i}^{(l)} \hat{\mathbf{m}}_{2,i}^{(l)}\right), \tag{8}$$

where $\rho(\cdot)$ is a nonlinear activation function. This adaptive fusion enables each node to dynamically combine simplicial features from different orders according to its local simplicial context.

Node update and classification. After adaptive gated fusion, AGSP updates the 0-simplex feature with the gated simplicial feature: $\mathbf{m}_{0,i}^{(l+1)} = \phi\left(\mathbf{m}_{0,i}^{(l)}, \hat{\mathbf{m}}_i^{(l)}\right)$, where $\phi(\cdot)$ denotes the node update function. After iterating through this process for L layers, the final node feature $\mathbf{m}_{0,i}^{(L)}$ is fed into an MLP classifier: $\hat{\mathbf{y}}_i = \mathrm{Softmax}\left(\mathrm{MLP}\left(\mathbf{m}_{0,i}^{(L)}\right)\right)$. AGSP is trained in an end-to-end manner by minimizing the cross-entropy loss. Algorithm 1 summarizes the overall procedure of the AGSP model.

4.2 Theoretical Analysis

In this section, we theoretically analyze the proposed AGSP algorithm.

Definition 1 (Permutation Equivariance on Simplicial Complexes) Let $\mathcal{K} = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2)$ be a 2-dimensional simplicial complex over the node set $\mathcal{V} = \{v_1, \ldots, v_N\}$, where $\mathcal{S}_0$, $\mathcal{S}_1$, and $\mathcal{S}_2$ denote the sets of 0-simplices, 1-simplices, and 2-simplices, respectively. We identify a vertex v_i with its corresponding 0-simplex $[v_i]$ when indexing node-level feature matrices.

Let $\pi : \mathcal{V} \mapsto \mathcal{V}$ be a permutation of the node set. For a k-simplex $\sigma = [v_{i_0}, \ldots, v_{i_k}] \in \mathcal{S}_k$, the action of π on σ is defined as $\pi(\sigma) = [\pi(v_{i_0}), \ldots, \pi(v_{i_k})]$. For each $k \in \{0, 1, 2\}$, the induced action on the set of k-simplices is $\pi(\mathcal{S}_k) = \{\pi(\sigma) : \sigma \in \mathcal{S}_k\}$. Therefore, the permuted simplicial complex is $\pi(\mathcal{K}) = (\pi(\mathcal{S}_0), \pi(\mathcal{S}_1), \pi(\mathcal{S}_2))$. Let $\mathbf{P}_\pi \in \{0,1\}^{N \times N}$ be the corresponding node permutation matrix, defined by $(\mathbf{P}_\pi)_{ab} = \begin{cases} 1, & v_a = \pi(v_b), \\ 0, & \text{otherwise.} \end{cases}$ If a simplex σ^k is represented by its node-incidence vector $\boldsymbol{\chi}_{\sigma^k} \in \{0,1\}^N$, then $\boldsymbol{\chi}_{\pi(\sigma^k)} = \mathbf{P}_\pi \boldsymbol{\chi}_{\sigma^k}$.

For each modality m, the action of π on the feature matrix is $\pi(\mathbf{X}^{(m)}) = \mathbf{P}_\pi \mathbf{X}^{(m)}$. A neural model $\mathcal{N} : \left(\mathcal{K}, \{\mathbf{X}^{(m)}\}_{m=1}^M\right) \mapsto \mathbf{H} \in \mathbb{R}^{N \times H}$ is defined on simplicial complexes, mapping the input simplicial complex and multimodal node features to a hidden node feature matrix $\mathbf{H}$, where H denotes the feature dimension of the final hidden layer. The model is permutation-equivariant if, for any node permutation π, $\mathcal{N}\left(\pi(\mathcal{K}), \{\mathbf{P}_\pi \mathbf{X}^{(m)}\}_{m=1}^M\right) = \mathbf{P}_\pi \mathcal{N}\left(\mathcal{K}, \{\mathbf{X}^{(m)}\}_{m=1}^M\right)$.

Definition 2 (Simplicial Awareness) Let $\mathcal{N}_\theta$ be a node-level neural model defined on 2-dimensional simplicial complexes $\mathcal{K} = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2)$ that satisfy downward closure. We focus on positive-dimensional simplices, i.e., $k \in \{1, 2\}$. For $k \in \{1, 2\}$, $\mathcal{N}_\theta$ is said to be k-simplex-aware if there exist two valid 2-dimensional simplicial complexes $\mathcal{K} = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2)$ and $\mathcal{K}' = (\mathcal{S}_0', \mathcal{S}_1', \mathcal{S}_2')$ such that the following two conditions hold: (1) the two complexes differ only in their k-simplex sets, i.e., $\mathcal{S}_j = \mathcal{S}_j', \forall j \in \{0, 1, 2\} \setminus \{k\}$ and $\mathcal{S}_k \neq \mathcal{S}_k'$; and (2) given the same multimodal node features $\{\mathbf{X}^{(m)}\}_{m=1}^M$, there exists a parameter setting θ under which the model produces different node-level outputs: $\mathcal{N}_\theta\left(\mathcal{K}, \{\mathbf{X}^{(m)}\}_{m=1}^M\right) \neq \mathcal{N}_\theta\left(\mathcal{K}', \{\mathbf{X}^{(m)}\}_{m=1}^M\right)$. If this property holds for every $k \in \{1, 2\}$, then $\mathcal{N}_\theta$ is simplicially aware of both 1-simplices and 2-simplices.

Lemma 1 (Face-invariant Simplex Feature Construction) For every 1-simplex $[v_i, v_j] \in \mathcal{S}_1$, its feature $\mathbf{m}_{1,ij}^{(l)}$ produced by AGSP is invariant to the ordering of its two vertices. For every 2-simplex $[v_i, v_j, v_k] \in \mathcal{S}_2$, its feature $\mathbf{m}_{2,ijk}^{(l)}$ produced by AGSP is invariant under any permutation of its three vertices (equivalently, under any permutation of its three boundary 1-faces). Consequently, AGSP constructs simplex features from their faces in a way that respects the unordered nature and downward-closure structure of simplicial complexes.

Proof. Please see Appendix B for the proof.

Theorem 1 (Permutation Equivariance of AGSP) For any fixed set of AGSP parameters, AGSP is permutation-equivariant in the sense of **Definition 1**. That is, for any permutation π over $\mathcal{V}$, $\mathbf{P}_\pi$ denotes the corresponding permutation matrix, and $\pi(\mathcal{K})$ denotes the simplicial complex obtained by applying π to every vertex of every simplex in $\mathcal{K}$. Then,

$$\mathrm{AGSP}\left(\pi(\mathcal{K}), \{\mathbf{P}_\pi \mathbf{X}^{(m)}\}_{m=1}^M\right) = \mathbf{P}_\pi \mathrm{AGSP}\left(\mathcal{K}, \{\mathbf{X}^{(m)}\}_{m=1}^M\right).$$

Proof. Please see Appendix C for the proof.

Theorem 2 (Simplicial Awareness of AGSP) The AGSP propagation layer is k-simplex-aware for $k \in \{1, 2\}$ in the simplicial complex sense of **Definition 2**. More precisely, there exist valid 2-dimensional simplicial complexes, shared multimodal node features, and non-degenerate AGSP parameter settings such that changing the k-simplex set changes the output of the AGSP propagation layer. Hence, AGSP is simplicially aware of both 1-simplices and 2-simplices.

Proof. Please see Appendix D for the proof.

Proposition 1 (Expressive Power beyond 1-WL-Bounded Pairwise MPNNs) There exist two non-isomorphic multimodal graphs with the same number of nodes and the same nonzero multimodal feature vector assigned to every node, such that any standard pairwise MPNN whose expressive power is bounded by 1-WL produces the same multiset of node features. In contrast, after clique-complex lifting, AGSP with a non-degenerate parameter setting can distinguish them through their induced 2-simplex structures.

Proof. Please see Appendix E for the proof.

Proposition 2 (Adaptive Gated Fusion Generalizes Fixed Weighted Fusion) Let $\mathbf{p}_i = \hat{\mathbf{m}}_{1,i}^{(l)}$ and $\mathbf{q}_i = \hat{\mathbf{m}}_{2,i}^{(l)}$ denote the multi-order topological messages of node v_i at layer l, respectively. We consider the fusion of these two messages and omit the shared node-level term $\hat{\mathbf{m}}_{0,i}^{(l)}$ and the subsequent activation since they are common to both fusion rules. Let $\mathcal{F}_{\text{fix}}$ be the function class realized by fixed weighted fusion $\mathbf{m}_i^{\text{fix}} = \lambda \mathbf{p}_i + (1-\lambda)\mathbf{q}_i$, where the coefficient $\lambda \in (0, 1)$ is shared by all nodes and is independent of the input. Let $\mathcal{F}_{\text{gate}}$ be the function class realized by adaptive gated fusion $\mathbf{m}_i^{\text{gate}} = g_{1,i}^{(l)} \mathbf{p}_i + g_{2,i}^{(l)} \mathbf{q}_i$, where $[g_{1,i}^{(l)}, g_{2,i}^{(l)}] = \text{Softmax}\left(\mathbf{W}_g[\hat{\mathbf{m}}_{0,i}^{(l)} \| \mathbf{p}_i \| \mathbf{q}_i] + \mathbf{b}_g\right)$. Then adaptive gated fusion strictly generalizes fixed weighted fusion: $\mathcal{F}_{\text{fix}} \subsetneq \mathcal{F}_{\text{gate}}$. In particular, normalized fixed summation corresponds to the special case $\lambda = \frac{1}{2}$.

Proof. Please see Appendix F for the proof.

Corollary 1 (Node-wise Structural Adaptivity) The adaptive gate in AGSP provides a node-dependent soft weighting mechanism between multi-order simplicial messages. Nodes with different local simplicial contexts can receive different gate coefficients, whereas fixed weighted fusion applies input-independent coefficients shared by all nodes.

Proof. Please see Appendix G for the proof.

4.3 Time Complexity

Let N_0, N_1, and N_2 denote the numbers of 0-simplices, 1-simplices, and 2-simplices, respectively. Here, L denotes the number of propagation layers, $D = \sum_{m=1}^{M} d_m$ denotes the concatenated multimodal input feature dimension, ν_i denotes the degree of node v_i in $\mathcal{G}$, and H denotes the feature dimension of the final hidden layer. Under clique-complex lifting, the construction of the simplicial complex costs $\mathcal{O}(N_0 + N_1 + \overline{\nu} N_1 + N_2)$ (Line 1), where $\overline{\nu} = \frac{1}{N_1} \sum_{(v_i, v_j) \in \mathcal{E}} \min\{\nu_i, \nu_j\}$. Initializing 0-simplex node features costs $\mathcal{O}(N_0 D)$ (Line 3). In each propagation layer, constructing the attention-based 1-simplex and 2-simplex features costs $\mathcal{O}(N_1 H^2)$ and $\mathcal{O}(N_2 H^2)$, respectively (Lines 6-7). Feature projection, adaptive gating, gated fusion, and node updates together cost $\mathcal{O}(N_0 H^2 + N_0 H)$ (Lines 8-11). The final MLP classifier costs $\mathcal{O}(N_0 H C)$ (Line 14). Therefore, the overall time complexity of AGSP is $\mathcal{O}\left(N_0 + N_1 + \overline{\nu} N_1 + N_2 + N_0 D + L\left((N_0 + N_1 + N_2) H^2 + N_0 H\right) + N_0 H C\right)$. The propagation term $\mathcal{O}\left(L(N_0 + N_1 + N_2) H^2\right)$ usually dominates the computational cost. Practically, H is usually set to 128 and is independent of graph size; therefore, the H^2 factor does not dominate the scalability of AGSP.

5 Experiments

In this section, we conduct experiments on four benchmark datasets: Grocery [40], Toys [40], Books [31], and Ele-fashion [31]. The details of these datasets are summarized in Table 2 of Appendix H. We further compare AGSP with several state-of-the-art baselines from three categories: traditional node classification methods in multimodal graphs (i.e., HGNN-IMA [35], UniGraph2 [1],

Table 1: Performance comparison on four multimodal graph datasets. Best results are in **bold**, and second results are underlined. OOM denotes out of memory.

Datasets	Metrics	HGNN-IMA	UniGraph2	NTSFormer	ZEN	CIF	RAGTrans	HiPoNet	CWN	MPS-Bot	AGSP
Grocery	Accuracy	84.23±0.49	74.82±0.79	85.40±0.21	72.86±0.70	85.98±0.40	85.00±0.20	82.90±0.30	77.15±1.07	67.05±1.29	**86.11±0.76**
	AUC	95.48±0.29	94.43±0.30	97.18±0.12	94.16±0.41	97.95±0.17	97.99±0.12	97.39±0.22	95.29±0.19	93.97±0.95	**98.03±0.08**
	F1	75.46±0.96	59.96±1.30	77.86±1.09	64.03±0.90	80.21±0.34	74.35±1.51	70.82±0.58	61.79±1.51	55.30±0.26	**84.76±0.64**
	Precision	77.67±1.27	62.27±1.39	78.63±0.87	63.74±0.92	82.05±0.64	75.10±1.64	74.45±1.18	63.20±1.73	57.45±0.39	**85.70±0.54**
	Recall	74.48±0.76	59.82±1.41	77.93±1.05	69.48±1.33	79.42±0.39	74.94±1.56	70.59±0.50	61.73±1.58	58.41±1.90	**84.93±0.64**
Toys	Accuracy	79.99±0.46	76.14±0.81	80.35±0.13	75.24±0.50	80.49±0.28	81.99±0.82	80.68±0.15	77.91±0.55	69.43±1.84	**83.19±1.05**
	AUC	96.16±0.42	95.77±0.23	97.77±0.10	95.73±0.12	97.82±0.06	98.06±0.10	97.73±0.06	96.64±0.07	94.80±0.87	**98.29±0.20**
	F1	77.03±0.59	71.49±0.90	78.35±0.46	71.18±0.32	78.23±0.23	78.63±1.41	76.57±0.33	73.89±0.55	53.98±1.99	**80.15±1.04**
	Precision	78.58±0.47	74.28±1.10	79.27±0.34	70.35±0.43	79.38±0.43	80.24±0.94	78.98±0.44	75.55±0.61	53.54±1.49	**81.63±1.00**
	Recall	76.33±0.47	70.50±0.89	78.10±0.42	73.34±0.47	77.79±0.39	78.03±1.66	76.49±0.45	73.34±0.70	56.92±1.14	**80.81±0.87**
Ele-fashion	Accuracy	85.94±0.14	78.16±0.14	87.04±0.03	72.29±0.45	84.32±0.10	86.14±0.26	86.00±0.11	85.16±0.13	76.92±1.49	**87.25±0.09**
	AUC	96.18±0.16	95.06±0.08	96.00±0.02	93.21±0.08	95.44±0.14	96.49±0.20	93.04±0.05	92.32±0.12	94.71±0.48	**96.82±0.04**
	F1	69.43±0.79	51.38±2.94	72.52±1.06	59.30±1.95	61.70±0.91	69.97±2.16	69.95±0.70	69.41±1.01	62.92±0.90	**73.40±0.13**
	Precision	**81.46±0.37**	62.97±4.70	72.23±0.75	57.10±1.87	70.66±1.17	75.44±1.23	79.96±2.95	75.48±0.33	65.09±0.99	76.04±0.15
	Recall	67.85±0.77	51.38±2.73	74.38±1.27	68.07±2.38	61.80±0.75	69.54±1.80	68.48±0.70	67.80±1.19	69.05±1.40	**74.94±0.21**
Books	Accuracy	71.69±0.76	74.73±0.05	71.76±0.13	OOM	62.01±0.48	OOM	71.18±0.06	71.78±0.48	66.85±0.35	**76.48±0.27**
	AUC	94.84±0.35	94.41±0.04	93.62±0.06	OOM	87.28±0.25	OOM	90.97±0.03	90.94±0.16	92.59±0.81	**95.52±0.57**
	F1	65.97±2.35	67.51±0.31	67.99±0.17	OOM	56.76±1.44	OOM	65.93±0.12	67.58±1.02	58.54±0.94	**70.63±0.52**
	Precision	73.90±0.09	64.35±0.65	69.82±0.19	OOM	62.18±0.67	OOM	73.41±0.14	72.68±0.70	50.15±1.07	**76.07±0.16**
	Recall	62.47±2.66	68.02±0.18	66.54±0.18	OOM	54.24±1.28	OOM	62.68±0.14	64.66±1.32	52.22±1.19	**68.31±0.75**

and NTSFormer [36]), hypergraph-based methods (i.e., ZEN [5], CIF [6], and RAGTrans [4]), and simplicial complex-based methods for unimodal graphs (i.e., HiPoNet [39], CWN [24], and MPS-Bot [41]). Details of the baselines and experimental settings can be found in Appendix I.

5.1 Results

Table 1 shows the experimental results. The proposed AGSP algorithm outperforms traditional multimodal graph node classification methods (i.e., HGNN-IMA, UniGraph2, and NTSFormer) in terms of accuracy, AUC, and F1 score, and it achieves the best precision in most cases. These methods mainly rely on pairwise interactions and thus fail to exploit higher-order relationships. By utilizing simplicial complexes, AGSP captures higher-order structural patterns overlooked by traditional methods. Furthermore, AGSP outperforms hypergraph-based methods (i.e., ZEN, CIF, and RAGTrans). Although these methods explicitly define higher-order relationships, they may struggle to capture potential hidden relations, especially in multimodal graphs. In contrast, AGSP leverages simplicial complexes to model higher-order relations implicit in graph topology, leading to improved performance for node classification in multimodal graphs. Additionally, while methods such as HiPoNet, CWN, and MPS-Bot use simplicial complexes, they focus on single-modal classification and are limited in handling multimodal data. AGSP is designed for multimodal graphs and integrates information from different modalities, demonstrating stronger classification capability.

A key observation is that the methods encountering out-of-memory (OOM) issues mainly rely on hypergraph structures. This suggests that explicitly defining higher-order relationships in hypergraphs may introduce heavy memory consumption on the largest dataset, i.e., Books. In contrast, AGSP uses simplicial complexes to incorporate higher-order structures with better memory efficiency, thus avoiding OOM issues. Additional visualization results in Appendix J further show that AGSP produces more compact intra-class clusters and clearer inter-class separation than representative baselines, which is consistent with the quantitative results.

5.2 Ablation Study

To evaluate the contributions of the key components in the AGSP framework, we conduct ablation studies on the Grocery dataset. Specifically, we construct two unimodal variants using only textual or visual features to evaluate the benefits of multimodal information. We then remove the simplicial construction to evaluate the role of higher-order relation modeling and replace the adaptive gating mechanism with several simplified fusion variants to assess the effectiveness of dynamic relation balancing.

As shown in Figure 3, the full AGSP achieves the best performance among all variants in terms of both AUC and F1. Using only textual or visual features leads to performance degradation compared with the full model, demonstrating that the two modalities provide complementary information and that jointly exploiting multimodal features benefits node classification. Removing the simplicial construction causes a clear performance drop, which indicates the importance of capturing higher-order relations beyond pairwise connections.

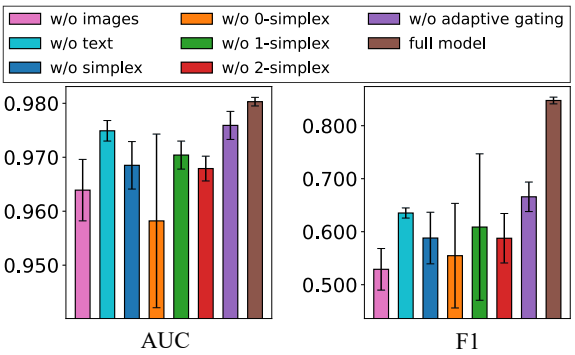


Figure 3: Ablation study of AGSP with different input modalities, simplex orders and fusion strategies.

In addition, replacing the adaptive gating mechanism with fixed fusion variants also degrades performance, which shows that dynamically balancing multi-order topological relations is crucial for node classification.

Overall, the ablation study verifies that multimodal information, explicit simplicial modeling, and adaptive gated fusion are important for the effectiveness of AGSP. Additional results on accuracy, precision, and recall are provided in Figure 7 of Appendix K.

5.3 Topological Analysis of Datasets

To examine whether the dataset topology supports the design of AGSP, we analyze the 2-simplex coverage of nodes and edges. Figure 4 shows that a substantial proportion of nodes and edges participate in at least one 2-simplex across all datasets, which indicates that higher-order structures are not rare but prevalent topological patterns across the datasets. Consequently, this observation justifies the construction of 2-simplices. Additional analysis on clustering coefficients and homophily within the 2-simplices is provided in Figure 8 of Appendix L.

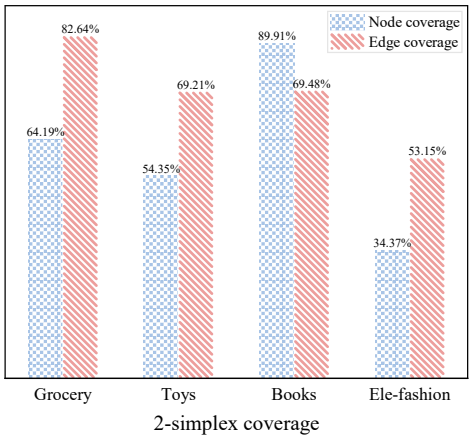


Figure 4: Topological analysis of node and edge participation in 2-simplices.

5.4 Hyper-parameter Sensitivity

We further perform hyper-parameter sensitivity analysis on the Grocery dataset with respect to dropout and learning rate, with the results shown in Figures 5. Specifically, the dropout ratio is varied from 0.1 to 0.5 in intervals of 0.1. The learning rate is set from 1×10^{-4} to 1×10^{-2}.

The results reveal noticeable trends in terms of AUC and F1. Increasing the learning rate from 1×10^{-4} to 1×10^{-3} generally improves performance, while further increasing it to 1×10^{-2} leads to clear performance degradation. This is likely because an excessively large learning rate makes the optimization process unstable, thereby preventing AGSP from converging to a good solution. In terms of dropout, AGSP achieves better performance under moderate regularization, with the best result obtained when the dropout ratio is 0.4 and the learning rate is 1×10^{-3}, corresponding to an AUC of 98.03% and an F1 score of 84.76%. When the dropout ratio is too small, the model may suffer from insufficient regularization; when the dropout ratio is

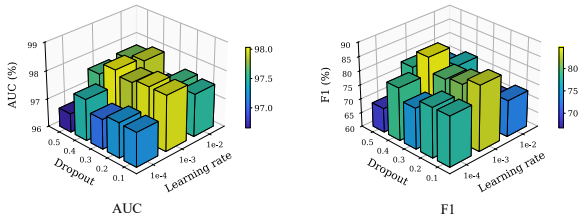


Figure 5: Sensitivity of AGSP to the dropout ratio and learning rate, evaluated by AUC and F1.

too large, excessive feature corruption may weaken useful multimodal information and higher-order simplicial structural signals, resulting in degraded node classification performance.

These observations suggest that balancing dropout and learning rate is critical for avoiding underfitting or overfitting, as well as for ensuring stable and effective training. Additional sensitivity results on accuracy, precision, and recall are provided in Figure 9 of Appendix M.

6 Conclusion

We propose AGSP, an end-to-end framework for node classification in multimodal graphs. By leveraging topology-induced higher-order simplicial structures and adaptive gated fusion, AGSP captures structural dependencies beyond pairwise propagation while dynamically balancing multi-order messages for each node. Extensive experiments on four real-world multimodal graph datasets demonstrate consistent improvements over competitive baselines, highlighting the effectiveness of simplicial higher-order modeling in multimodal graphs.

Acknowledgments

This work was supported by Open Project of Xiangjiang Laboratory (No.25XJ03020), National Natural Science Foundation of China (No. 62302528, 62672517, 62172449, 72374070), Hunan Provincial Natural Science Foundation of China (2025JJ20071), the Independent Research Project of National Key Laboratory of Big Data and Decision, and the High Performance Computing Center of Central South University.

References

[1] Y. He, Y. Sui, X. He, Y. Liu, Y. Sun, and B. Hooi, "Unigraph2: Learning a Unified Embedding Space to Bind Multimodal Graphs," in *WWW*, 2025, pp. 1759–1770.

[2] X. Ning, D. Fu, T. Wei, W. Xu, and J. He, "Graph4mm: Weaving Multimodal Learning with Structural Information," in *ICML*, 2025.

[3] Y. Ektefaie, G. Dasoulas, A. Noori, M. Farhat, and M. Zitnik, "Multimodal Learning with Graphs," *Nature Machine Intelligence*, vol. 5, no. 4, pp. 340–350, 2023.

[4] Z. Cheng, J. Zhang, X. Xu, G. Trajcevski, T. Zhong, and F. Zhou, "Retrieval-Augmented Hypergraph for Multimodal Social Media Popularity Prediction," in *KDD*, 2024, pp. 445–455.

[5] C. Bae, D. Choi, J. Lee, and J. Yoo, "Parameter-Free Hypergraph Neural Network for Few-Shot Node Classification," in *NeurIPS*, 2025.

[6] Y. Lin, H. Yan, X. Tong, Y. Chang, H. Wang, Z. Zhou, S. Gao, Y. Wang, and W. Zhang, "Commonality in Few: Few-Shot Multimodal Anomaly Detection via Hypergraph-Enhanced Memory," in *AAAI*, vol. 40, no. 9, 2026, pp. 7015–7023.

[7] W. Wu, S. Li, L. Chen, F. Li, and C. Luo, "Sketching Very Large-Scale Dynamic Attributed Networks More Practically," in *WWW*, 2025, pp. 5264–5274.

[8] W. Wu, S. Li, M. Jiang, C. Luo, and F. Li, "Time-and Space-Efficiently Sketching Billion-Scale Attributed Networks," *IEEE Transactions on Knowledge and Data Engineering*, pp. 966–978, 2024.

[9] Z. Wang, Z. Zhang, T. Ma, N. V. Chawla, C. Zhang, and Y. Ye, "Beyond Message Passing: Neural Graph Pattern Machine," in *ICML*, 2025.

[10] Y. Wang, M. Yasunaga, H. Ren, S. Wada, and J. Leskovec, "Vqa-Gnn: Reasoning with Multimodal Knowledge via Graph Neural Networks for Visual Question Answering," in *ICCV*, 2023, pp. 21 582–21 592.

[11] H. Liu, C. Li, and L. Tian, "Multi-Modal Graph Attention Network for Video Recommendation," in *CCET*, 2022, pp. 94–99.

[12] J. Chang, C. Gao, X. He, D. Jin, and Y. Li, "Bundle Recommendation with Graph Convolutional Networks," in *SIGIR*, 2020, pp. 1673–1676.

[13] Y. Li, C. Gao, H. Luo, D. Jin, and Y. Li, "Enhancing Hypergraph Neural Networks with Intent Disentanglement for Session-Based Recommendation," in *SIGIR*, 2022, pp. 1997–2002.

[14] G. Xv, C. Lin, W. Guan, J. Gou, X. Li, H. Deng, J. Xu, and B. Zheng, "E-Commerce Search via Content Collaborative Graph Neural Network," in *KDD*, 2023, pp. 2885–2897.

[15] W. Zhou, T. Li, P. Vougiouklis, M. Steedman, and J. Z. Pan, "A Usage-Centric Take on Intent Understanding in E-Commerce," in *EMNLP*, 2024, pp. 228–236.

[16] Q. F. Lotito, F. Musciotto, A. Montresor, and F. Battiston, "Higher-Order Motif Analysis in Hypergraphs," *Communications Physics*, vol. 5, no. 1, p. 79, 2022.

[17] J. Jo, J. Baek, S. Lee, D. Kim, M. Kang, and S. J. Hwang, "Edge Representation Learning with Hypergraphs," in *NeurIPS*, 2021, pp. 7534–7546.

[18] X. Sun, H. Yin, B. Liu, H. Chen, J. Cao, Y. Shao, and N. Q. Viet Hung, "Heterogeneous Hypergraph Embedding for Graph Classification," in *WSDM*, 2021, p. 725733.

[19] W. Ju, Z. Mao, S. Yi, Y. Qin, Y. Gu, Z. Xiao, Y. Wang, X. Luo, and M. Zhang, "Hypergraph-Enhanced Dual Semi-Supervised Graph Classification," in *ICML*, 2024, pp. 22 594–22 604.

[20] Y. Feng, J. Han, S. Ying, and Y. Gao, "Hypergraph Isomorphism Computation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 3880–3896, 2024.

[21] M. Hayhoe, H. M. Riess, M. M. Zavlanos, V. Preciado, and A. Ribeiro, "Transferable Hypergraph Neural Networks via Spectral Similarity," in *LoG*, 2024, pp. 18–1.

[22] Y. Zhang, M. Lucas, and F. Battiston, "Higher-Order Interactions Shape Collective Dynamics Differently in Hypergraphs and Simplicial Complexes," *Nature Communications*, vol. 14, no. 1, p. 1605, 2023.

[23] R. Yang, F. Sala, and P. Bogdan, "Efficient Representation Learning for Higher-Order Data with Simplicial Complexes," in *LoG*, 2022, pp. 13–1.

[24] C. Bodnar, F. Frasca, Y. Wang, N. Otter, G. F. Montufar, P. Lio, and M. Bronstein, "Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks," in *ICML*, 2021, pp. 1026–1037.

[25] S. Gurugubelli and S. P. Chepuri, "SaNN: Simple Yet Powerful Simplicial-aware Neural Networks," in *ICLR*, 2024.

[26] K. Maggs, C. Hacker, and B. Rieck, "Simplicial Representation Learning with Neural k-Forms," in *ICLR*, 2024.

[27] C. Zhou, X. Wang, and M. Zhang, "Facilitating Graph Neural Networks with Random Walk on Simplicial Complexes," in *NeurIPS*, 2023, pp. 16 172–16 206.

[28] H. Wu, A. Yip, J. Long, J. Zhang, and M. K. Ng, "Simplicial Complex Neural Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 1, pp. 561–575, 2024.

[29] X. Tan, W. Wu, and C. Luo, "SCHash: Speedy Simplicial Complex Neural Networks via Randomized Hashing," in *SIGIR*, 2023, pp. 1609–1618.

[30] W. Wu, X. Tan, Y. Peng, L. Chen, F. Li, and C. Luo, "Beyond Node-Centric Modeling: Sketching Signed Networks with Simplicial Complexes," in *NeurIPS*, 2025.

[31] J. Zhu, Y. Zhou, S. Qian, Z. He, T. Zhao, N. Shah, and D. Koutra, "Mosaic of Modalities: A Comprehensive Benchmark for Multimodal Graph Learning," in *CVPR*, 2025, pp. 14 215–14 224.

[32] W. Wu, Y. Peng, L. Chen, X. Tan, J. Yang, Z. Wang, F. Li, and C. Luo, "Learning-Free Continuous-Attribute Graph Embedding via Locality-Sensitive Hashing," *Knowledge-Based Systems*, p. 116482, 2026.

[33] Y. Wei, X. Wang, L. Nie, X. He, R. Hong, and T.-S. Chua, "Mmgcn: Multi-Modal Graph Convolution Network for Personalized Recommendation of Micro-Video," in *ACM MM*, 2019, pp. 1437–1445.

[34] Z. Tao, Y. Wei, X. Wang, X. He, X. Huang, and T.-S. Chua, "Mgat: Multimodal Graph Attention Network for Recommendation," *Information Processing & Management*, vol. 57, no. 5, p. 102277, 2020.

[35] J. Li, J. Zhu, L. Chang, Y. Li, M. Li, Y. Wang, Y. Yang, and H. Wang, "Representation Learning with Mutual Influence of Modalities for Node Classification in Multi-Modal Heterogeneous Networks," in *IJCAI*, 2025.

[36] J. Hu, Y. He, Y. Li, B. Hooi, and B. He, "Ntsformer: A Self-Teaching Graph Transformer for Multimodal Isolated Cold-Start Node Classification," in *AAAI*, vol. 40, no. 17, 2026, pp. 14 856–14 864.

[37] M. Hajij, G. Zamzmi, T. Papamarkou, N. Miolane, A. Guzm'an-S'aenz, K. N. Ramamurthy, T. Birdal, T. Dey, S. Mukherjee, S. N. Samaga, N. Livesay, R. Walters, P. Rosen, and M. T. Schaub, "Topological Deep Learning: Going Beyond Graph Data," in *NeurIPS*, 2022.

[38] S. Kim, S. Y. Lee, Y. Gao, A. Antelmi, M. Polato, and K. Shin, "A Survey on Hypergraph Neural Networks: An In-Depth and Step-by-Step Guide," in *KDD*, 2024, pp. 6534–6544.

[39] S. Viswanath, H. Madhu, D. Bhaskar, J. Kovalic, D. R. Johnson, C. Tape, I. Adelstein, R. Ying, M. Perlmutter, and S. Krishnaswamy, "Hiponet: A Multi-View Simplicial Complex Network for High Dimensional Point-Cloud and Single-Cell Data," in *NeurIPS*, 2025.

[40] H. Yan, C. Li, J. Yin, Z. Yu, W. Han, M. Li, Z. Zeng, H. Sun, and S. Wang, "When Graph Meets Multimodal: Benchmarking and Meditating on Multimodal Attributed Graph Learning," in *KDD*, 2025, pp. 5842–5853.

[41] F. Li, H. Zhang, X. Zhang, and W. Wu, "Unmasking Bots in Higher Dimensions: Message Passing over Simplexes for Bot Detection," in *WWW*, 2026, p. 47134721.

[42] C. Bodnar, F. Frasca, N. Otter, Y. G. Wang, P. Liò, G. Montúfar, and M. Bronstein, "Weisfeiler and Lehman Go Cellular: Cw Networks," in *NeurIPS*, 2021, pp. 2625–2640.

[43] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How Powerful are Graph Neural Networks?" in *ICLR*, 2019.

Appendix

A Simplicial Complex

For a simplex $\sigma^k = [v_0, v_1, \ldots, v_k]$, any non-empty subset $\tau \subseteq \sigma^k$ is called a face of σ^k. If $\dim(\tau) = p < k$, then τ is called a p-face of σ^k. The $(k-1)$-faces of σ^k are obtained by removing one vertex, i.e., $\sigma^k_{-r} = [v_0, \ldots, v_{r-1}, v_{r+1}, \ldots, v_k]$ for $r \in \{0, \ldots, k\}$. For example, the boundary 1-faces of a 2-simplex $[v_i, v_j, v_k]$ are $[v_i, v_j]$, $[v_i, v_k]$, and $[v_j, v_k]$.

A simplicial complex $\mathcal{K}$ over $\mathcal{V}$ is a collection of simplices satisfying the downward-closure property, i.e., if $\sigma \in \mathcal{K}$ and $\emptyset \neq \tau \subseteq \sigma$, then $\tau \in \mathcal{K}$. This property means that simplicial complexes provide a downward-closed higher-order structure: whenever a higher-order simplex is included, all of its lower-order faces are also included. For instance, $[v_i, v_j, v_k] \in \mathcal{K}$ implies $[v_i, v_j], [v_i, v_k], [v_j, v_k] \in \mathcal{K}$ and $[v_i], [v_j], [v_k] \in \mathcal{K}$. However, the converse does not necessarily hold: the existence of all three edges $[v_i, v_j]$, $[v_i, v_k]$, and $[v_j, v_k]$ does not imply that the filled 2-simplex $[v_i, v_j, v_k]$ must be included in a simplicial complex. This distinction is important because two simplicial complexes can share the same 1-skeleton, i.e., the same vertices and edges [42], but differ in their higher-order simplex sets.

In graph $\mathcal{G}$, we construct the simplicial complex $\mathcal{K}$ by clique-complex lifting [24]. Specifically, the 0-simplex set is defined as $\mathcal{S}_0 = \{[v_i] : v_i \in \mathcal{V}\}$. The 1-simplex set is defined as $\mathcal{S}_1 = \{[v_i, v_j] : (v_i, v_j) \in \mathcal{E}\}$. The 2-simplex set is defined as $\mathcal{S}_2 = \{[v_i, v_j, v_k] : (v_i, v_j), (v_i, v_k), (v_j, v_k) \in \mathcal{E}\}$. Thus, every 3-clique in the input graph is lifted to a filled triangle, i.e., a 2-simplex. In this way, the original graph structure is enriched with topology-induced higher-order relations among triples of mutually connected nodes.

For each node v_i, we denote its incident 1-simplex and 2-simplex by $\mathcal{I}_1(i) = \{\sigma \in \mathcal{S}_1 : v_i \in \sigma\}$ and $\mathcal{I}_2(i) = \{\tau \in \mathcal{S}_2 : v_i \in \tau\}$, respectively. Here, $\mathcal{I}_1(i)$ contains all edges incident to v_i, while $\mathcal{I}_2(i)$ contains all filled triangles that include v_i. These incident simplex sets are used to aggregate higher-order simplicial messages for each node.

B Face-invariant Simplex Feature Construction

Proof. For any 1-simplex $[v_i, v_j] \in \mathcal{S}_1$, AGSP defines its feature as

$$\mathbf{m}^{(l)}_{1,ij} = \mathbf{m}^{(l)}_{0,i} \odot \mathbf{m}^{(l)}_{0,j}.$$

Since the Hadamard product is commutative,

$$\mathbf{m}^{(l)}_{0,i} \odot \mathbf{m}^{(l)}_{0,j} = \mathbf{m}^{(l)}_{0,j} \odot \mathbf{m}^{(l)}_{0,i},$$

we have

$$\mathbf{m}^{(l)}_{1,ij} = \mathbf{m}^{(l)}_{1,ji}.$$

Therefore, the 1-simplex feature does not depend on the arbitrary ordering of its two vertices.

For any 2-simplex $[v_i, v_j, v_k] \in \mathcal{S}_2$, AGSP defines its feature as

$$\mathbf{m}^{(l)}_{2,ijk} = \mathrm{MaxPool}\{\mathbf{m}^{(l)}_{1,ij}, \mathbf{m}^{(l)}_{1,ik}, \mathbf{m}^{(l)}_{1,jk}\}.$$

A permutation of the vertices of $[v_i, v_j, v_k]$ only permutes its three boundary 1-faces, i.e., $[v_i, v_j], [v_i, v_k]$ and $[v_j, v_k]$. Since $\mathrm{MaxPool}$ is a symmetric set operation, permuting these boundary 1-simplex features does not change the resulting 2-simplex feature. Hence, $\mathbf{m}^{(l)}_{2,ijk}$ depends only on the set of boundary 1-faces of $[v_i, v_j, v_k]$, not on their order. This proves that the AGSP simplex feature construction is face-invariant. □

C Permutation Equivariance of AGSP

Proof. We prove the claim by induction over the propagation layers.

First, we clarify the induced action of a node permutation on simplices. A node permutation $\pi : \mathcal{V} \mapsto \mathcal{V}$ is defined on the node set. It naturally induces a relabeling action on simplices by applying

π to each of their vertices: $\pi(v_0, \ldots, v_k) = (\pi(v_0), \ldots, \pi(v_k))$. Accordingly, for each $k \in \{0, 1, 2\}$, we define $\pi(\mathcal{S}_k) = \{\pi(\sigma^k) : \sigma^k \in \mathcal{S}_k\}$. Furthermore, we use $\pi(\mathcal{K})$ to denote the simplicial complex obtained by relabeling every vertex in every simplex based on π: $\pi(\mathcal{K}) = (\pi(\mathcal{S}_0), \pi(\mathcal{S}_1), \pi(\mathcal{S}_2))$.

Let $\mathbf{M}_{\mathcal{K}}^{(l)}$ denote the node feature matrix at layer l when AGSP is applied to the simplicial complex $\mathcal{K}$. According to the 0-simplex initialization in Eq. (1), AGSP constructs the initial node features by concatenating the textual and visual features:

$$\mathbf{M}_{\mathcal{K}}^{(0)} = [\mathbf{X}^{(1)} \| \mathbf{X}^{(2)}].$$

When the input is permuted by π, the textual and visual feature matrices become $\mathbf{P}_\pi \mathbf{X}^{(1)}$ and $\mathbf{P}_\pi \mathbf{X}^{(2)}$, respectively. Therefore, the initial node feature matrix of the permuted input is

$$\mathbf{M}_{\pi(\mathcal{K})}^{(0)} = [\mathbf{P}_\pi \mathbf{X}^{(1)} \| \mathbf{P}_\pi \mathbf{X}^{(2)}].$$

Since $\mathbf{P}_\pi$ permutes rows and feature concatenation is performed along the feature dimension, we have

$$[\mathbf{P}_\pi \mathbf{X}^{(1)} \| \mathbf{P}_\pi \mathbf{X}^{(2)}] = \mathbf{P}_\pi [\mathbf{X}^{(1)} \| \mathbf{X}^{(2)}].$$

Hence,

$$\mathbf{M}_{\pi(\mathcal{K})}^{(0)} = \mathbf{P}_\pi \mathbf{M}_{\mathcal{K}}^{(0)}.$$

Thus, the equivariance relation holds for $l = 0$.

We assume that the equivariance relation holds at layer l, i.e.,

$$\mathbf{M}_{\pi(\mathcal{K})}^{(l)} = \mathbf{P}_\pi \mathbf{M}_{\mathcal{K}}^{(l)}.$$

For each node v_i, let $v_{i'} = \pi(v_i)$ denote its relabeled counterpart in the permuted complex. Then, the induction hypothesis can be written as

$$\mathbf{m}_{0,i'}^{(l)}(\pi(\mathcal{K})) = \mathbf{m}_{0,i}^{(l)}(\mathcal{K}).$$

It remains to prove that this relation is preserved after the next AGSP propagation layer as follows.

For any 1-simplex $[v_i, v_j] \in \mathcal{S}_1$, its permuted counterpart is $[\pi(v_i), \pi(v_j)] \in \pi(\mathcal{S}_1)$. By the induction hypothesis,

$$\mathbf{m}_{0,\pi(i)}^{(l)}(\pi(\mathcal{K})) = \mathbf{m}_{0,i}^{(l)}(\mathcal{K}), \qquad \mathbf{m}_{0,\pi(j)}^{(l)}(\pi(\mathcal{K})) = \mathbf{m}_{0,j}^{(l)}(\mathcal{K}).$$

Therefore, the corresponding 1-simplex interaction is preserved under permutation:

$$\mathbf{m}_{1,\pi(i)\pi(j)}^{(l)}(\pi(\mathcal{K})) = \mathbf{m}_{1,ij}^{(l)}(\mathcal{K}).$$

By **Lemma 1**, this relation is independent of the ordering of the two vertices.

Similarly, for any 2-simplex $[v_i, v_j, v_k] \in \mathcal{S}_2$, its permuted counterpart is $[\pi(v_i), \pi(v_j), \pi(v_k)] \in \pi(\mathcal{S}_2)$. Since the corresponding boundary 1-simplex interactions are preserved under the induction hypothesis and the 2-simplex feature construction is invariant to the ordering of its boundary faces, we have

$$\mathbf{m}_{2,\pi(i)\pi(j)\pi(k)}^{(l)}(\pi(\mathcal{K})) = \mathbf{m}_{2,ijk}^{(l)}(\mathcal{K}).$$

Moreover, the incident 1-simplices and 2-simplices of v_i are mapped exactly to the incident 1-simplices and 2-simplices of $\pi(v_i)$. Since AGSP uses shared attention parameters, the same attention mechanism is applied after permutation. The summation over incident simplex sets is also preserved. Therefore, the attention weights and the aggregated messages are preserved under permutation:

$$\mathbf{m}_{1,\pi(i)}^{(l)}(\pi(\mathcal{K})) = \mathbf{m}_{1,i}^{(l)}(\mathcal{K}), \qquad \mathbf{m}_{2,\pi(i)}^{(l)}(\pi(\mathcal{K})) = \mathbf{m}_{2,i}^{(l)}(\mathcal{K}).$$

The projection matrices $\mathbf{W}_0, \mathbf{W}_1, \mathbf{W}_2$, the gate parameters $\mathbf{W}_g, \mathbf{b}_g$, the nonlinear activation function, and the node update function are shared across nodes. Hence, the projected messages, gate coefficients, and fused simplicial features also satisfy

$$\hat{\mathbf{m}}_{\pi(i)}^{(l)}(\pi(\mathcal{K})) = \hat{\mathbf{m}}_i^{(l)}(\mathcal{K}).$$

Together with the induction hypothesis, the node update gives

$$\mathbf{m}_{0,\pi(i)}^{(l+1)}(\pi(\mathcal{K})) = \mathbf{m}_{0,i}^{(l+1)}(\mathcal{K}).$$

This node-wise relation can be written in matrix form as

$$\mathbf{M}_{\pi(\mathcal{K})}^{(l+1)} = \mathbf{P}_\pi \mathbf{M}_{\mathcal{K}}^{(l+1)}.$$

By induction, the final node features and the prediction logits are permutation-equivariant. Hence,

$$\mathrm{AGSP}\left(\pi(\mathcal{K}), \{\mathbf{P}_\pi \mathbf{X}^{(m)}\}_{m=1}^{M}\right) = \mathbf{P}_\pi \mathrm{AGSP}\left(\mathcal{K}, \{\mathbf{X}^{(m)}\}_{m=1}^{M}\right).$$

□

D Simplicial Awareness of AGSP

Proof. We prove the claim separately for 1-simplex awareness and 2-simplex awareness. The following constructions are made in the simplicial-complex setting, where the simplex sets only need to satisfy downward closure.

1-simplex awareness. We consider two valid 2-dimensional simplicial complexes on the same 0-simplex set

$$\mathcal{S}_0 = \{[v_1], [v_2], [v_3], [v_4]\}.$$

Let

$$\mathcal{S}_1 = \{[v_1, v_2], [v_3, v_4]\}, \qquad \mathcal{S}_1' = \{[v_3, v_4]\}, \qquad \mathcal{S}_2 = \mathcal{S}_2' = \emptyset.$$

The corresponding simplicial complexes are defined as

$$\mathcal{K} = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2), \qquad \mathcal{K}' = (\mathcal{S}_0, \mathcal{S}_1', \mathcal{S}_2').$$

These two complexes have identical 0-simplex and 2-simplex sets but different 1-simplex sets. Both are valid simplicial complexes because every 1-simplex has its boundary 0-faces contained in $\mathcal{S}_0$, and no 2-simplex is present.

Let both complexes be equipped with the same multimodal node features $\{\mathbf{X}^{(m)}\}_{m=1}^{M}$. Since node v_1 is incident to the 1-simplex $[v_1, v_2]$ in $\mathcal{K}$, it receives the pairwise simplicial message

$$\mathbf{m}_{1,12}^{(l)} = \mathbf{m}_{0,1}^{(l)} \odot \mathbf{m}_{0,2}^{(l)}.$$

In $\mathcal{K}'$, node v_1 has no incident 1-simplex, so the corresponding aggregated 1-simplex message is

$$\mathbf{m}_{1,1}^{(l)}(\mathcal{K}') = \mathbf{0}.$$

For suitable shared node features and a non-degenerate incident-simplex aggregator,

$$\mathbf{m}_{1,1}^{(l)}(\mathcal{K}) \neq \mathbf{m}_{1,1}^{(l)}(\mathcal{K}').$$

Since $\mathbf{m}_{1,1}^{(l)}(\mathcal{K}) \neq \mathbf{m}_{1,1}^{(l)}(\mathcal{K}')$, we can choose parameters so that the 1-simplex branch preserves this difference. Specifically, $\mathbf{W}_1$ does not map the nonzero 1-simplex message to zero, while the gate assigns a positive weight $g_{1,1}^{(l)} > 0$ to this branch. Since the 0-simplex and 2-simplex parts are identical in the two complexes, the only difference in the fusion input of v_1 comes from the 1-simplex branch. Under the above non-degenerate parameter setting, this difference is preserved after projection and then assigned a positive gate weight. Therefore, the fused feature of v_1 differs between $\mathcal{K}$ and $\mathcal{K}'$, and the updated features of v_1 differ as well. Finally, AGSP is aware of 1-simplices.

2-simplex awareness. For 2-simplex awareness, two valid 2-dimensional simplicial complexes are constructed on the same 0-simplex set

$$\mathcal{S}_0 = \{[v_1], [v_2], [v_3]\}$$

and with the same 1-simplex set

$$\mathcal{S}_1 = \mathcal{S}_1' = \{[v_1, v_2], [v_1, v_3], [v_2, v_3]\}.$$

Let

$$\mathcal{S}_2 = \{[v_1, v_2, v_3]\}, \qquad \mathcal{S}_2' = \emptyset.$$

The two simplicial complexes are defined as

$$\mathcal{K} = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2), \qquad \mathcal{K}' = (\mathcal{S}_0, \mathcal{S}_1', \mathcal{S}_2').$$

Both are valid simplicial complexes. In $\mathcal{K}$, the included 2-simplex $[v_1, v_2, v_3]$ satisfies downward closure because all its non-empty faces are present. In $\mathcal{K}'$, the three 1-faces exist without the filled 2-simplex, which is also valid because downward closure does not require the converse.

In $\mathcal{K}$, the 2-simplex feature is

$$\mathbf{m}_{2,123}^{(l)} = \mathrm{MaxPool}\{\mathbf{m}_{1,12}^{(l)}, \mathbf{m}_{1,13}^{(l)}, \mathbf{m}_{1,23}^{(l)}\}.$$

Thus, each node in $\{v_1, v_2, v_3\}$ receives a higher-order message through its incident 2-simplex. In $\mathcal{K}'$, no 2-simplex exists, so

$$\mathbf{m}_{2,i}^{(l)}(\mathcal{K}') = \mathbf{0}, \qquad v_i \in \{v_1, v_2, v_3\}.$$

For suitable shared node features and a non-degenerate incident-simplex aggregator,

$$\mathbf{m}_{2,i}^{(l)}(\mathcal{K}) \neq \mathbf{m}_{2,i}^{(l)}(\mathcal{K}'), \qquad v_i \in \{v_1, v_2, v_3\}.$$

Since $\mathbf{W}_2$ is non-degenerate and the softmax gate gives $g_{2,i}^{(l)} > 0$, the 2-simplex branch contributes differently to the updates in $\mathcal{K}$ and $\mathcal{K}'$. Under a non-degenerate parameter setting in which the subsequent shared operations do not collapse this difference, the node features produced by AGSP differ between the two complexes. Hence, AGSP is aware of 2-simplex.

Therefore, AGSP is aware of both 1-simplices and 2-simplices. □

Remark on Clique-complex Lifting. **Theorem 2** analyzes the sensitivity of the AGSP propagation layer to the simplicial-complex structure. It does not imply that, under a fixed clique-complex lifting rule, one can arbitrarily include or remove a 2-simplex while keeping the same ordinary graph. In AGSP, once the input graph is fixed, clique-complex lifting deterministically determines $\mathcal{S}_2$. Therefore, for ordinary graph inputs, the higher-order advantage of AGSP comes from exploiting the 2-simplex structures deterministically induced by clique-complex lifting. This is consistent with **Proposition 1**, where different ordinary graphs induce different 2-simplex structures after lifting.

E Expressive Power beyond 1-WL-Bounded Pairwise MPNNs

Proof. We construct two multimodal graphs with the same number of nodes and the same nonzero multimodal feature vector assigned to every node. Let $\mathcal{G}_1$ be the complete bipartite graph $K_{3,3}$, and let $\mathcal{G}_2$ be the triangular prism graph. Both graphs are 3-regular, with six nodes.

Since all nodes are initialized with identical multimodal features, all nodes receive the same initial 1-WL label. Moreover, because both $\mathcal{G}_1$ and $\mathcal{G}_2$ are 3-regular, each node observes the same multiset of neighbor labels at every iteration. Therefore, the 1-WL refinement assigns the same label to all nodes in both graphs throughout the refinement process [43].

We then apply the clique-complex lifting used by AGSP. Since $K_{3,3}$ is bipartite, it contains no 3-cliques. Hence, its lifted simplicial complex has

$$\mathcal{S}_2(\mathcal{G}_1) = \emptyset.$$

By contrast, the triangular prism graph contains two 3-cliques, which are lifted to two 2-simplices under clique-complex lifting. Therefore,

$$\mathcal{S}_2(\mathcal{G}_2) \neq \emptyset.$$

For every node in $\mathcal{G}_1$, the incident 2-simplex set is empty, i.e., $\mathcal{I}_2(i) = \emptyset$, and thus the aggregated 2-simplex message is

$$\mathbf{m}_{2,i}^{(l)}(\mathcal{G}_1) = \mathbf{0}.$$

For every node in $\mathcal{G}_2$, the node is incident to one of the two lifted 2-simplices, so $\mathcal{I}_2(i) \neq \emptyset$. With suitable nonzero multimodal node features and non-degenerate AGSP parameters, the incident 2-simplex aggregation produces a nonzero message:

$$\mathbf{m}_{2,i}^{(l)}(\mathcal{G}_2) \neq \mathbf{0}.$$

The gate coefficient $g_{2,i}^{(l)}$ is strictly positive due to softmax normalization. In addition, the 2-simplex projection and subsequent shared transformations are assumed not to collapse this difference. Therefore, the AGSP update in $\mathbf{m}_{0,i}^{(l+1)} = \phi\left(\mathbf{m}_{0,i}^{(l)}, \hat{\mathbf{m}}_i^{(l)}\right)$, produces different node features for the lifted complexes of $\mathcal{G}_1$ and $\mathcal{G}_2$.

Thus, AGSP distinguishes this pair through the induced 2-simplex branch, whereas any standard 1-WL-bounded pairwise MPNN cannot distinguish them from identical multimodal node features. This establishes an expressive separation from standard pairwise message passing. □

Proposition 1 should be interpreted as a comparison with ordinary pairwise MPNNs, whose expressiveness is bounded by 1-WL. It does not claim that AGSP is more expressive than all possible graph neural architectures, such as models equipped with unique node identifiers, positional encodings, subgraph features, or higher-order transformers. Instead, it shows that the topology-induced 2-simplex branch gives AGSP access to structural information that is invisible to standard pairwise message passing.

F Adaptive Gated Fusion Generalizes Fixed Weighted Fusion

Proof. We first show that fixed weighted fusion is contained within the adaptive gated formulation. For any fixed coefficient $\lambda \in (0, 1)$, the gate parameters are set as $\mathbf{W}_g = \mathbf{0}$, with the gate bias chosen as

$$\mathbf{b}_g = [\log \lambda, \log(1 - \lambda)].$$

Then, for every node v_i,

$$[g_{1,i}^{(l)}, g_{2,i}^{(l)}] = \mathrm{Softmax}(\mathbf{b}_g) = [\lambda, 1 - \lambda].$$

It follows that

$$\mathbf{m}_i^{\mathrm{gate}} = \lambda \mathbf{p}_i + (1 - \lambda)\mathbf{q}_i = \mathbf{m}_i^{\mathrm{fix}}.$$

Hence,

$$\mathcal{F}_{\mathrm{fix}} \subseteq \mathcal{F}_{\mathrm{gate}}.$$

We next show that the inclusion is strict. It is sufficient to construct one gated fusion function that cannot be represented by any fixed weighted fusion rule. Since the vector-valued fusion is based on scalar-weighted addition, the scalar case can be obtained by restricting the input vectors to a one-dimensional coordinate subspace. Specifically, one coordinate r is fixed, and all vectors are restricted to have nonzero values only on this coordinate, i.e., $\mathbf{p}_i = p\mathbf{e}_r$, $\mathbf{q}_i = q\mathbf{e}_r$. The shared node-level input is set to zero, where $\mathbf{e}_r$ is the r-th standard basis vector. Under this restriction, the vector-valued gated fusion is reduced to a scalar function on the r-th coordinate, denoted as $m^{\mathrm{gate}}(p, q)$. Let the gate parameters be chosen such that $\exists a \neq 0$,

$$g_1^{(l)}(p, q) = \frac{\exp(ap)}{\exp(ap) + 1}, \qquad g_2^{(l)}(p, q) = \frac{1}{\exp(ap) + 1}.$$

Such a gate can be realized by selecting the coordinate p in the first logit, with the second logit set to zero.

For inputs with $q = 0$, the gated fusion becomes

$$m^{\mathrm{gate}}(p, 0) = \frac{\exp(ap)}{\exp(ap) + 1} p.$$

If this function can be represented by a fixed weighted fusion rule, then there exists a constant $\lambda \in (0, 1)$ such that $\forall p$, this implies

$$\frac{\exp(ap)}{\exp(ap) + 1} p = \lambda p.$$

Table 2: Statistics of the datasets.

Dataset	#Nodes	#Edges	#Classes	#Triangles	Avg. Degree
Grocery	17,074	171,340	20	125,689	20.07
Toys	20,695	126,886	18	55,690	12.26
Ele-fashion	97,766	199,602	12	131,458	4.08
Books	685,294	7,235,084	11	7,466,373	21.12

However, when $a \neq 0$, the function $\frac{\exp(ap)}{\exp(ap)+1}$ is non-constant with respect to p, whereas λ is input-independent. Therefore, no fixed coefficient λ can satisfy the above equality for all p.

Hence,

$$\mathcal{F}_{\text{fix}} \subsetneq \mathcal{F}_{\text{gate}}.$$

□

G Node-wise Structural Adaptivity

Proof. The adaptive gate in AGSP is computed from the node-specific gate input

$$[\hat{\mathbf{m}}_{0,i}^{(l)} \| \hat{\mathbf{m}}_{1,i}^{(l)} \| \hat{\mathbf{m}}_{2,i}^{(l)}],$$

which contains multi-order topological messages of node v_i. Therefore, when two nodes have different gate inputs and the gate parameters are non-degenerate, their gate coefficients $(g_{1,i}^{(l)}, g_{2,i}^{(l)})$ may differ according to their multimodal features and local simplicial contexts.

The softmax normalization guarantees

$$g_{1,i}^{(l)}, g_{2,i}^{(l)} \geq 0, \qquad g_{1,i}^{(l)} + g_{2,i}^{(l)} = 1.$$

Hence, the gated part of the fused message is a node-dependent convex combination:

$$g_{1,i}^{(l)} \hat{\mathbf{m}}_{1,i}^{(l)} + g_{2,i}^{(l)} \hat{\mathbf{m}}_{2,i}^{(l)}.$$

The essential property is not merely convexity, but node dependence. For one node, AGSP may assign a larger coefficient to the 1-simplex message; for another node, it may assign a larger coefficient to the higher-order 2-simplex message. This allows AGSP to adapt to local topology since the availability and contribution of 2-simplex-based higher-order information may vary across nodes.

In contrast, fixed weighted fusion uses input-independent coefficients shared by all nodes, for example

$$\lambda \hat{\mathbf{m}}_{1,i}^{(l)} + (1-\lambda) \hat{\mathbf{m}}_{2,i}^{(l)}, \qquad \lambda \in (0,1).$$

Since the same coefficient λ is applied to every node, fixed weighted fusion cannot provide such node-wise structural calibration. This proves the claimed node-wise structural adaptivity of AGSP. □

H Datasets

We evaluate our method on four real-world multimodal attributed graphs, namely Grocery, Toys, Books, and Ele-fashion. The first two datasets are taken from MAGB [40], while the latter two are adopted from MM-Graph [31]. Each node is associated with both textual and visual attributes.

- Grocery [40] is an e-commerce graph collected from the Amazon "Grocery" domain. Nodes correspond to products, edges represent frequent co-view or co-purchase relations, and each node is associated with textual and visual product information.
- Toys [40] is an e-commerce graph constructed from the Amazon "Toys & Games" domain, where each node denotes a product and each edge indicates that two products are frequently co-viewed or co-purchased. The textual attributes are given by product titles and descriptions, and the visual attributes are provided by product images.

- Ele-fashion [31] is a fashion-product graph based on the Amazon-Fashion dataset, where each node corresponds to a product and each edge captures a co-purchase relation between two products. The textual attributes are given by product titles, and the visual attributes are provided by product images.
- Books [31] is a book graph based on the Goodreads Book Graph dataset, where each node represents a book and an edge indicates that a user who likes one book is also likely to like the other. The textual attributes are constructed from book descriptions, and the visual attributes are given by book covers.

For all four datasets, the provided category labels are used as node classes. The raw text-image pairs are treated as multimodal node attributes. The statistics of the datasets are summarized in Table 2.

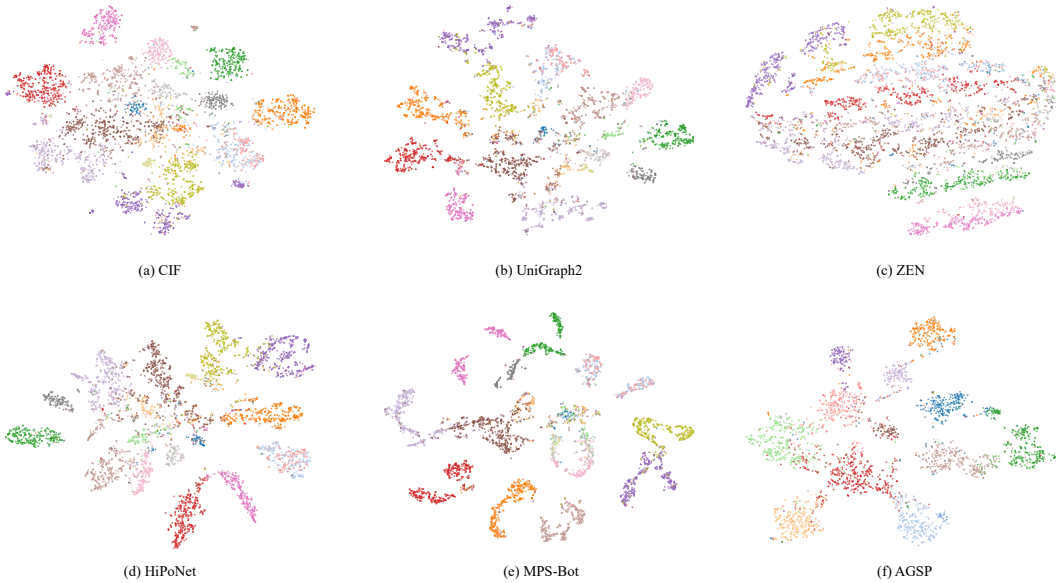
I Experimental Setting

We compare AGSP with baselines from three categories, including traditional node classification methods in multimodal graphs, hypergraph-based methods, and simplicial complex-based methods for unimodal graphs. For all compared methods, the official implementations released by the original authors are used, and the recommended hyperparameter settings are followed.

1. **Traditional node classification methods in multimodal graphs.** These methods mainly perform multimodal learning and message passing over pairwise graph structures.
   - **HGNN-IMA** [35] learns node classification in multimodal graphs by modeling the mutual influence across different modalities.
   - **UniGraph2** [1] addresses node classification in multimodal graphs by aligning and integrating multimodal information within a unified graph learning framework.
   - **NTSFormer** [36] is a self-teaching graph transformer designed for multimodal isolated cold-start node classification.
2. **Hypergraph-based methods.** These methods model higher-order relations through explicitly defined hyperedges.
   - **ZEN** [5] is a parameter-free hypergraph neural network for few-shot node classification that exploits higher-order relations.
   - **CIF** [6] introduces a hypergraph-enhanced memory mechanism to capture common patterns in few-shot multimodal settings.
   - **RAGTrans** [4] employs retrieval-augmented hypergraph modeling for multimodal node classification.
3. **Simplicial complex-based methods for unimodal graphs.** These methods exploit simplicial structures to capture higher-order relations for classification tasks, but they are designed for unimodal graph settings.
   - **HiPoNet** [39] is a multi-view simplicial complex network that captures higher-order topological structures in structured data.
   - **CWN** [24] performs message passing on simplicial complexes to model interactions beyond pairwise relations.
   - **MPS-Bot** [41] leverages message passing over simplices to exploit higher-dimensional interactions for node classification.

To ensure a fair comparison, we evaluate all methods under the same data split, preprocessing pipeline, and evaluation protocol. For methods originally designed for classification, we retain their original training objectives and output layers. For methods originally proposed for other predictive tasks, we keep their original backbone architectures unchanged and attach the same lightweight classification head to adapt them to our classification setting. In this way, the comparison focuses on the node classification ability of each backbone rather than on task-specific output designs.

Unless otherwise specified, all model-specific settings are inherited from the official implementations. For all methods, the input data, training/validation/testing splits, and evaluation metrics are kept identical. We report the average results of accuracy, AUC, F1, precision, and recall over five runs. All experiments are conducted on a Linux server equipped with 128 Intel Xeon Gold 6248



(a) CIF (b) UniGraph2 (c) ZEN

(d) HiPoNet (e) MPS-Bot (f) AGSP

Figure 6: t-SNE visualization of the final hidden features before the classifier on the Toys dataset. Different colors denote different classes.

CPUs, 600 GB RAM, and dual NVIDIA Tesla V100S (32 GB) GPUs. When out-of-memory errors occur on the GPU, we switch the computation to the CPU. If the method still exceeds the available memory budget or cannot finish within a reasonable runtime, we report it as OOM.

J Visualization

To further examine the ability of AGSP, we visualize the node embeddings extracted from the last hidden layer before the classifier using t-SNE, as shown in Figure 6. Different colors denote different node classes. For clarity, we select five representative baselines from the three categories discussed above: UniGraph2 from multimodal graph learning methods, CIF and ZEN from hypergraph-based methods, and HiPoNet and MPS-Bot from simplicial complex-based methods. These baselines are selected because they cover the main competing paradigms for multimodal graph learning and higher-order graph learning while keeping the visualization readable.

As shown in Figure 6, AGSP produces more compact intra-class clusters and clearer inter-class separation than the baselines. Most node classes form distinguishable groups with less overlap, indicating that the hidden features learned by AGSP are more discriminative for node classification. This advantage mainly stems from the simplicial propagation layer, which captures higher-order relations beyond pairwise connections, and the adaptive gating mechanism, which dynamically balances multi-order topological relations for each node. These observations are consistent with the quantitative results in Table 1, further verifying the effectiveness of AGSP for node classification in multimodal graphs.

By contrast, the representative baselines exhibit different degrees of class overlap and cluster fragmentation. UniGraph2 can form several local clusters, but samples from different classes are still mixed in some central regions. This may be because traditional multimodal graph learning methods mainly rely on pairwise message passing, which limits their ability to capture higher-order relational patterns hidden in the graph topology. CIF and ZEN explicitly introduce hypergraph-based modeling to exploit higher-order relations, but their clusters remain scattered or partially overlapped. A possible reason is that hypergraph-based methods usually depend on predefined or constructed hyperedges, which may not fully reveal topology-induced latent higher-order relations in multimodal graphs. HiPoNet and MPS-Bot achieve better local grouping than the above methods, showing the benefit of simplicial complex modeling. However, their clusters are still fragmented, and some classes remain close to each other. This is likely because these methods are mainly designed for uni-

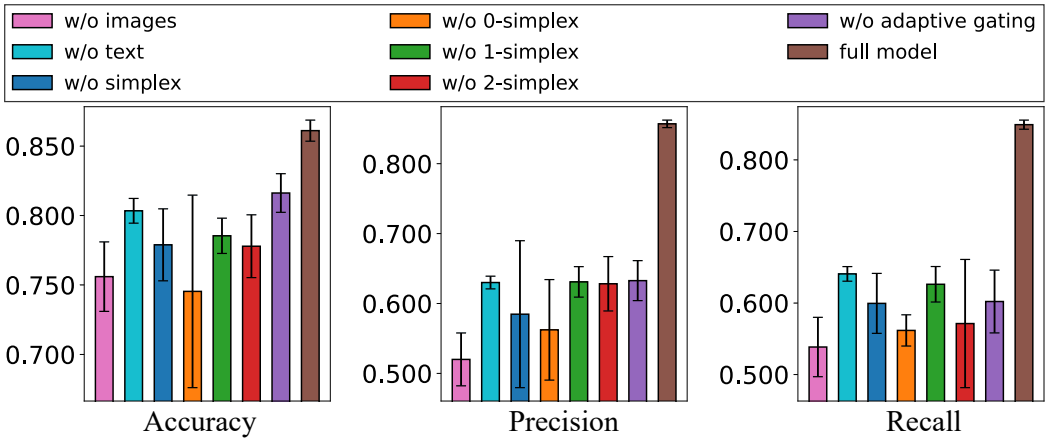


Figure 7: Ablation study of the proposed model. We compare the full model with variants using different simplex orders and a variant without adaptive gating. Results are reported in terms of accuracy, precision and recall.

modal graph learning, which limits their ability to exploit complementary information from multiple modalities.

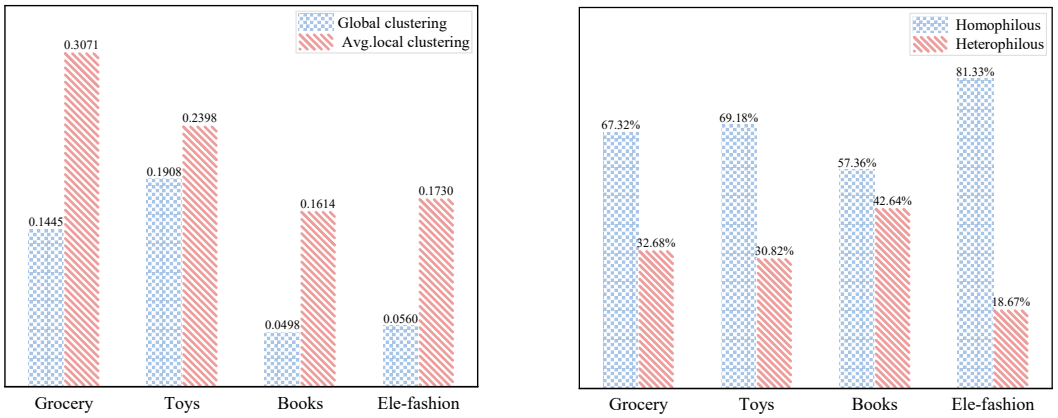
In summary, AGSP generates more compact and better separated clusters. The intra-class samples are more tightly grouped, while different classes are more clearly separated. This suggests that AGSP can better integrate multimodal node attributes with topology-induced higher-order simplicial relations. Moreover, the adaptive gating mechanism enables AGSP to dynamically balance multi-order topological relations for each node, leading to more discriminative node features for node classification.

K Ablation Study

As shown in Figure 7, the full AGSP consistently achieves the best performance in terms of accuracy, precision, and recall. Using only textual or visual features results in performance degradation across these metrics, further demonstrating the benefit of jointly exploiting complementary multimodal information. Removing the simplicial construction also leads to clear degradation, confirming the importance of capturing higher-order relations beyond pairwise connections. In addition, replacing the adaptive gating mechanism with simplified fusion variants reduces overall performance, indicating the effectiveness of dynamically balancing multi-order topological relations for node classification.

L Topological Analysis of Datasets

Figure 8(a) further reports the global clustering coefficient and the average local clustering coefficient. The global clustering coefficient measures the overall fraction of closed triplets in the graph, while the average local clustering coefficient averages the neighborhood closure ratio over individual nodes. Since closed triplets can be directly lifted to 2-simplices under clique-complex lifting, the clear gap between these two coefficients suggests that 2-simplices are unevenly distributed across nodes and local graph regions. This observation indicates that the usefulness of higher-order information is unlikely to be uniform across nodes, thereby supporting the need for adaptive gated fusion. In addition, Figure 8(b) further analyzes 2-simplex homophily. Specifically, a 2-simplex is considered homophilous if all its vertices share the same class label, and heterophilous otherwise. The results show that homophilous 2-simplices dominate heterophilous ones in all four datasets, indicating that these higher-order structures are strongly correlated with class semantics and can provide label-relevant information beyond pairwise relations. Overall, these observations further support the proposed AGSP architecture.



(a) Clustering coefficients

(b) Homophily in 2-simplex

Figure 8: Topological analysis of the four datasets. Clustering coefficients report the global clustering coefficient and the average local clustering coefficient. Homophily in 2-simplex compares the proportions of homophilous 2-simplices and heterophilous 2-simplices.

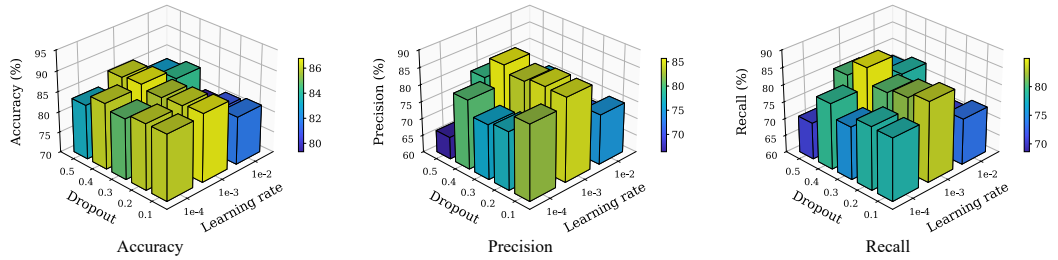


Figure 9: Additional sensitivity results of AGSP to the dropout ratio and learning rate, evaluated by Accuracy, Precision, and Recall.

M Hyper-parameter Sensitivity

Figure 9 reports additional sensitivity results in terms of accuracy, precision, and recall. Overall, AGSP shows a similar trend across these metrics. The performance generally improves when the learning rate increases from 10^{-4} to 10^{-3}, while a further increase to 10^{-2} leads to performance degradation. This again indicates that an overly large learning rate may make the optimization unstable.

In terms of dropout, AGSP achieves better results under moderate regularization. Extremely small dropout may provide insufficient regularization, while overly large dropout can weaken useful multi-modal and higher-order structural signals. The best or near-best results are usually obtained when the dropout ratio is 0.4 with a learning rate of 10^{-3}. These observations are consistent with the AUC and F1 results in the main text, which further demonstrate that AGSP is stable within a practical range of hyper-parameter settings.

N Limitations

This work has two main limitations. First, AGSP relies on topology-induced simplicial structures to model higher-order relations; thus, its performance may be affected by the quality and completeness of the input graph topology. Noisy or sparse graphs may result in incomplete or unreliable higher-order relational patterns. Second, the current framework focuses on simplicial relations up to dimension 2. Therefore, it does not explicitly preserve higher-dimensional simplicial interactions. Extending AGSP to incorporate higher-dimensional simplices could further enhance its ability to model richer group-wise dependencies, but this would increase computational complexity.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contents of the paper match the claims made in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations in a separate section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We have provided the theories and the corresponding proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We explicitly reveal all the hyperparameters and open source models necessary to reproduce our results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have provided the publicly available datasets and our code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so No is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We have provided the details of all the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report the mean and standard deviation over five independent runs for all main experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have described the system on which the experiments are conducted in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We review and ensure that the present work respects the NeurIPS Code of Ethics at each individual part.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The research does not have concerns about societal impacts because it is designed for general-purpose graph embedding.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We use publicly available datasets and models, widely used in the research community.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have obeyed all license and terms involved in the datasets and the compared methods.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve IRB approvals.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We just adopt LLM to proofread the paper (e.g., grammar, spelling, word choice).

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.