

Generative-Contrastive Heterogeneous Graph Neural Network

Yu Wang, Lei Sang*, Yi Zhang, Yiwen Zhang and Xindong Wu *Fellow, IEEE*

Abstract—Heterogeneous Graphs (HG) effectively model complex relationships in the real world through multi-type nodes and edges. In recent years, inspired by self-supervised learning (SSL), contrastive learning (CL)-based Heterogeneous Graphs Neural Networks (HGNNs) have shown great potential in utilizing data augmentation and contrastive discriminators for downstream tasks. However, data augmentation remains limited due to the graph data’s integrity. Furthermore, the contrastive discriminators suffer from sampling bias and lack local heterogeneous information. To tackle the above limitations, we propose a novel *Generative-Contrastive Heterogeneous Graph Neural Network (GC-HGNN)*. Specifically, we propose a heterogeneous graph generative learning method that enhances CL-based paradigm. This paradigm includes: 1) A contrastive view augmentation strategy using a masked autoencoder. 2) Position-aware and semantics-aware positive sample sampling strategy for generating hard negative samples. 3) A hierarchical contrastive learning strategy aimed at capturing local and global information. Furthermore, the hierarchical contrastive learning and sampling strategies aim to constitute an enhanced contrastive discriminator under the generative-contrastive perspective. Finally, we compare our model with seventeen baselines on eight real-world datasets. Our model outperforms the latest baselines on node classification and link prediction tasks. To reproduce our work, we have open-sourced our code at <https://github.com/wangyu0627/GC-HGNN>.

Index Terms—Self-Supervised Learning, Heterogeneous Graph Neural Networks, Contrastive Learning, Generative Learning.

I. INTRODUCTION

HETEROGENEOUS Information Networks (HIN) or Heterogeneous Graphs (HG) [1], [2] have emerged as critical tools for analyzing the real world, as they capture rich semantic information by modeling different types of nodes and edges. This is particularly relevant in networks such as citation networks [2], [3], recommender systems [4], [5], and social networks [6]. Recently, Heterogeneous Graph Neural Networks (HGNNs) [2], [7] demonstrate a strong ability in processing HIN data because they expand the receptive field of nodes through meta-path [8] and message-passing mechanism, thereby capturing the rich semantics.

Self-supervised learning (SSL) has been widely applied in the fields of Computer Vision (CV) [9], [10] and Natural Language Processing (NLP) [11], [12] leveraging unlabeled data

Yu Wang, Lei Sang, Yi Zhang and Yiwen Zhang, are with School of Computer Science and Technology, Anhui University 230601, Hefei, Anhui, China. E-mail: sanglei@ahu.edu.cn, wangyuhu@stu.ahu.edu.cn, zhangyi@stu.ahu.edu.cn, zhangyiwen@ahu.edu.cn.

Xindong Wu is with the Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei University of Technology, Hefei 230601, Anhui, P.R. China. E-mail: xwu@hfut.edu.cn

*Corresponding author.

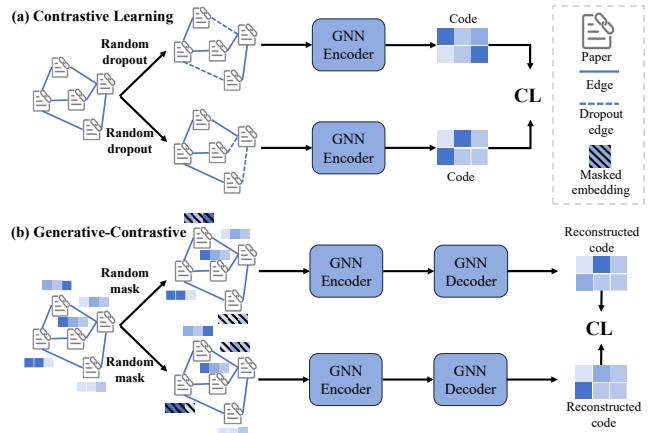


Fig. 1: (a) Contrastive Learning paradigm aligns two similar views constructed through random edge dropping; (b) Generative-Contrastive paradigm aligns two reconstructed embeddings without altering the original graph structure.

to learn informative representations. Inspired by this trend, researchers explore the combination of HGNNs and SSL. As a major paradigm, HGNNs [7], [13] based on Contrastive Learning (CL) show great potential, primarily due to their suitability for downstream tasks achieved by setting contrastive views and discriminators, as demonstrated in [9], [14]. Data augmentation [15], [16] serves as a tool to construct contrastive views, maximizing mutual information by aligning two views. However, given the requirement to maintain graph data’s integrity and unique features, it is challenging to provide reasonable data augmentation to leverage the benefits of contrastive learning.

Generative learning [11], [12] effectively addresses this limitation by reconstructing the data distribution, disregarding downstream tasks and data augmentation, thereby enhancing each node embedding. This idea further serves as a foundation for proposing a generative learning enhanced contrastive learning paradigm for heterogeneous graphs. As shown in Figure 1, first, HGNNs utilize a meta-path to obtain a graph containing nodes of the same type. CL-based HGNN [13], [17] generates supervised signals by aligning two similar views constructed through random edge dropping; meanwhile, generative-contrastive learning [18], [19] reconstructs the intrinsic structures and features of data by training an autoencoder and uses a contrastive discriminator to distinguish them in the embedding space. Despite the obvious advantages of the generative-contrastive paradigm, it presents several challenges for heterogeneous graphs:

CH1: Unbalanced data augmentation. The current graph augmentation strategies [15], [17], such as structural and feature augmentation, still have limitations. On the one hand, structural augmentation [13], [17], which involves randomly dropping out nodes and edges, might disrupt the graph integrity, as nodes and edges are the fundamental properties of a graph. On the other hand, feature augmentation [7], [20], which perturbs node attributes, tends to overlook the unique features of each node’s input. In these methods, certain nodes have more edges than others, which results in unbalanced supervised signals. Exploring how to design enhanced contrastive views represents a challenge that requires attention.

CH2: Negative sampling bias. In heterogeneous graphs, contrastive learning [7] heavily relies on high-quality negative samples. Moreover, the contrastive discriminator requires stricter sampling methods under the generative-contrastive paradigm [19]. Existing sampling strategies [7], [17] are typically based on randomness or meta-paths, and these approaches frequently lead to an increase in false negative samples. Recent works [21], [22] aim to select or synthesize hard negatives to mitigate the impact of false negatives. However, they are not suitable for HGs and provide limited benefits. Therefore, there is an urgent need to design an enhanced sampling strategy for HGs.

CH3: Lack local heterogeneous information. Existing methods [17], [23] rely solely on meta-paths to capture rich semantic information in HGs and often overlook the influence of one-hop neighbors on target nodes. Network schema leverages direct connections between nodes to capture the local structure of target nodes. In contrast, meta-paths are employed to extract higher-order structures. Designing an effective self-supervised mechanism that considers both local and global information can significantly enhance the model’s performance.

To address the three challenges mentioned above, we propose a heterogeneous graph self-supervised model called Generative-Contrastive Heterogeneous Graph Neural Network (GC-HGNN). For **CH1**, GC-HGNN employs a generative masked autoencoder (MAE) to enhance contrastive views. The MAE is capable of reconstructing node embeddings without altering the original graph structure and features. The masking strategy averages the edges of each node, ensuring that each node receives the same supervised signals during the reconstruction process. Moreover, random structural perturbations in existing methods can exacerbate imbalance, potentially leading to adverse effects on node representations [16]. Our proposed GC-HGNN uses an autoencoder to create embeddings without modifying the graph’s topology or inherent node features, preserving useful information. The proposed sampling strategy is not simply “node drop” but a selective uniform random sampling without replacement. This method helps prevent bias by ensuring neighbors are neither fully masked nor entirely visible, which enhances feature restoration [24]. For **CH2**, the model introduces a sampling strategy that integrates both location-awareness and semantic-awareness. This approach adaptively samples nodes based on the rich semantics inherent to HGs and the global positions of nodes within these graphs. In contrastive learning, the goal is to minimize positive sample distances while maximizing negative ones. In GC-HGNN,

the positive sampling strategy also improves hard negative selection, promoting feature similarity in positive pairs while reducing similarity in negative pairs. Studies show that sub-optimal false negatives can lead to erroneous guidance [25], and precise positive sampling enhances discrimination, thus indirectly supporting negative sample quality [26]. Finally, for **CH3**, GC-HGNN employs hierarchical contrastive learning (HCL) to capture both one-hop and higher-order neighbors. Intra-contrast is employed to contrast different meta-path views, while inter-contrast is used to contrast between network schema and meta-path views. Our model builds upon HeCo’s framework but simplifies and generalizes it by encoding all edge types directly, which minimizes preprocessing cost and avoids semantic loss. Unlike HeCo’s two-step attention aggregation, our approach is more streamlined.

We summarize the contributions of this paper as follows:

- We propose a generative learning method that enhances contrastive learning (CL)-based paradigm GC-HGNN, leveraging a generative masked autoencoder for view augmentation. This represents a significant exploration for graph generative-contrastive networks.
- We introduce hierarchical contrastive learning (HCL) as an enhanced contrastive discriminator, which is integrated with innovative sampling strategy to align the generative model. These modules effectively mine self-supervised signals.
- We conduct comprehensive experiments to evaluate GC-HGNN with seventeen baseline methods and eight real datasets. The experimental results demonstrate that GC-HGNN is superior to baselines and effectively extracts heterogeneous information.

RELATED WORK

This section examines the development process of heterogeneous graph neural networks. Here, we introduce the issue of infeasible and expensive data labels, further expanding on the self-supervised advantages and providing a brief overview of existing models.

A. Heterogeneous Graph Neural Networks

In recent years, Graph Neural Networks (GNNs) [27]–[30] have demonstrated significant success in fields such as social network analysis and recommender systems. This success is attributed to their effective modeling of graph structure information and the integration of both local and global information. These graph structures typically have different types of nodes and edges. Heterogeneous graph neural networks (HGNNs) are designed to better handle heterogeneous information.

The currently prevalent HGNN-based models [7], [31]–[34] can be categorized into meta-path-based methods and type-based methods. The classic meta-path-based models include HAN [2], HetGNN [32], and MAGNN [33]. Specifically, HAN employs a dual-layer attention mechanism at both the node and semantic levels to adaptively capture the relationships between nodes and meta-paths. HetGNN introduces a heterogeneous graph attention propagation mechanism, combining meta-path information and attention mechanisms to learn node representations. MAGNN integrates meta-paths and adaptive

mechanisms to learn node representations in heterogeneous graphs by learning meta-path attention weights at both the node and global levels. The type-based models include RGCN [35], HGT [36], and RHGNN [34]. RGCN captures rich semantic relationships by using different weight matrices for different types of edges. HGT achieves efficient representation learning and information propagation on heterogeneous graph data by introducing a multi-head attention mechanism and a heterogeneous graph structure encoder. RHGNN utilizes dedicated graph convolution components and cross-relation passing modules to learn node representations on heterogeneous graphs. However, the methods mentioned above focus exclusively on supervised heterogeneous graph models. FrameERC [37], [38] leverages the Framelet Transform in a multimodal GNN for emotion recognition in conversations. In order to deeply explore the potential relationships and patterns in the data and to effectively utilize unlabeled data, an increasing number of studies are beginning to focus on self-supervised learning methods.

B. Self-supervised Graph Learning

Given the rapid development of self-supervised learning in computer vision [9], [39] and natural language processing [11], [12] fields, graph neural networks have benefited from this progress. Self-supervised graph neural networks typically train without labels by generating contrastive views or reconstructing node attributes, thereby better leveraging the rich structural information in graph data. Transitioning from homogeneous graphs to heterogeneous graphs entails assigning different types or attributes to nodes and edges in the graph, rendering the graph structure more complex and diverse in representation.

Self-supervised heterogeneous graph neural networks can be categorized into contrastive and generative. Contrastive learning models [7], [17], [40] generate enhanced contrastive views and aim to maximize the mutual information between these views to produce supervised signals. Specifically, HDGI [40] utilizes graph convolutional modules and semantic-level attention to learn node representations and maximize local-global mutual information. HeCo [7] complements across perspectives through network schema and meta-path views, establishing a sampling contrast method between the two views to achieve self-supervised enhancement. HGCLM [17] solely utilizes the rich semantic relationships brought by meta-paths for graph enhancement, introducing multi-view contrastive learning. It also proposes sampling strategies for positive samples in both topology and semantics to alleviate sampling bias. Generative models [23], [41] commonly adopt graph autoencoder and aim to reconstruct features to obtain supervision. For example, MvdGAE [42] utilizes random edge dropout to enhance views and autoencoders to enforce the recovery of views for more robust representations. Meanwhile, HGMAE [23] further introduces a dynamic masking mechanism and position-aware encoding to enhance the model's performance and utilizes attribute and view dual-recovery to improve denoising capability.

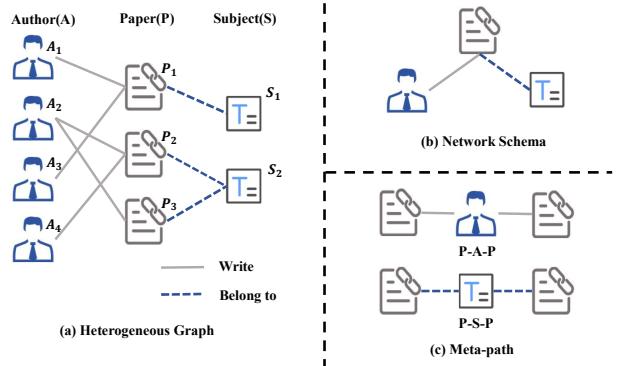


Fig. 2: A toy example of an HIN on ACM dataset.

TABLE I: Summary of symbols and notations.

Symbol	Description
\mathcal{G}	A heterogeneous information network (HIN)
V, E	Sets of nodes and edges
I, J	Sets of node types and edge types
T_G	Network schema of the HIN
ρ_j	A meta-path in the set of selected meta-paths
P_r	Projection matrix
α_{ew}	Attention weight between node e and neighbor w
σ	Activation function
\mathbf{h}_e^1	Updated embedding of node e in network schema
$G = \mathcal{V}, \mathcal{A}, \mathbf{X}$	Definition of a meta-path-based-subgraph with nodes, adjacency matrix, and feature matrix
$\mathcal{V}^{[Mask]}$	Set of masked nodes
$\frac{p}{\tilde{\mathbf{X}}}$	Mask ratio for masked nodes
$\mathcal{V}^{[Remask]}$	Masked node feature matrix
$\mathbf{H}, \widetilde{\mathbf{H}}, \mathbf{H}_{Re}$	Set of remasked nodes
w_{ρ_j}	Encoded feature, decoder input with the remasked feature matrix, reconstructed output
γ_{ρ_j}	Weight score for meta-path ρ_j
\mathbf{h}_e^2	Attention weight for meta-path ρ_j
\mathcal{E}	Updated embedding of node e in meta-path view
$\widetilde{\mathcal{V}}$	Edge sampling set for inter-contrast
$\mathbb{P}_i, \mathbb{N}_i$	Positive sample set for intra-contrast
τ	Positive and negative sample sets for inter-contrast
λ	Temperature parameter in contrastive loss
$\mathcal{L}_{Gen/Intra/Inter}$	Balance coefficient of contrastive loss \mathcal{L}_{Inter}
\mathcal{L}_{bpr}	Loss functions of generative learning, intra-contrast and inter-contrast
	Bayesian personalized ranking loss

II. PRELIMINARIES

In this section, we formally define some significant concepts related to HIN as follows:

Heterogeneous Information Network (HIN): A HIN is characterized by a graph structure $\mathcal{G} = (V, E, I, J, \phi, \varphi)$, in which the collections of nodes and edges are symbolized by V and E respectively. For every node v and edge e , there are associated type mapping functions $\phi : V \rightarrow I$ for nodes and $\varphi : E \rightarrow J$ for edges. Here, I represents the node types and J denotes the edge types, with the total number of types exceeding two, i.e., $|I| + |J| > 2$.

Network Schema: The network schema, denoted as $T_G = (I, J)$, is used to demonstrate the high-level topology of an HIN, which describes the node types and their interaction relations. The network schema itself is a directed graph that is predicated on a collection of node types I , where the links between them are represented by the set of relational edge

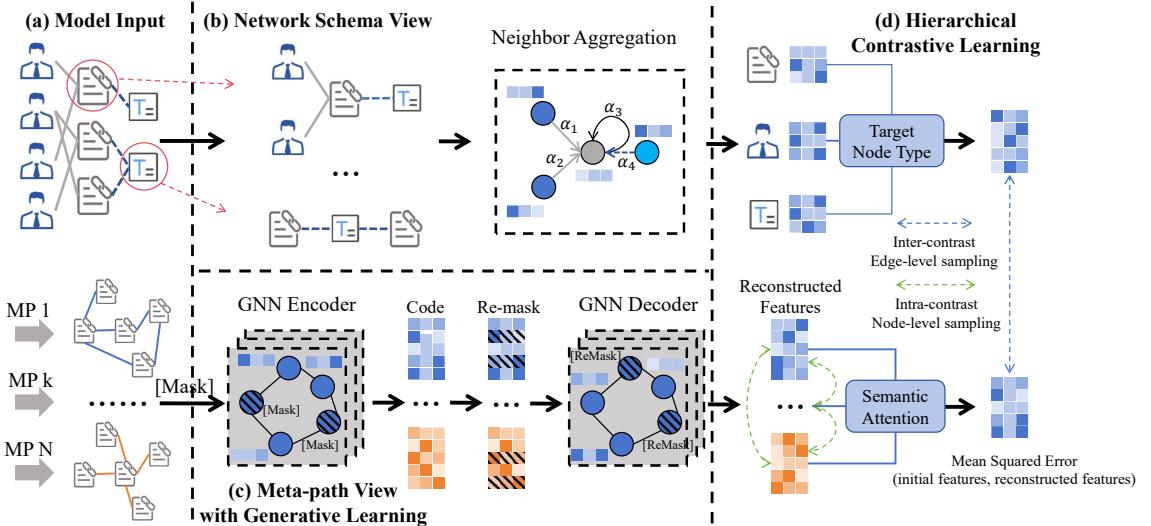


Fig. 3: The training framework of our proposed GC-HGNN. (a) The model respectively takes the entire heterogeneous graph and the views based on meta-paths, dividing them into inputs for the network schema View and the meta-path view; (b) the network schema view employs an attention mechanism to focus on one-hop neighbors of each node; (c) the meta-path view utilizes a masked autoencoder to generate views for intra-contrast; (d) the hierarchical contrast uses a proposed dynamic sampling mechanism to generate more hard negative samples to enhance the contrastive discriminator.

types J . The network schema for the ACM is depicted in the top portion of Figure 2 (b).

Meta-path: In an HIN $\mathcal{G} = (V, E, I, J, \phi, \varphi)$, a meta-path ρ is depicted as $I_1 \xrightarrow{J_1} I_2 \xrightarrow{J_2} \dots \xrightarrow{J_l} I_{l+1}$, which describes a composite connection between I_1 and I_{l+1} . Taking Figure 2 as an instance, multiple meta-paths, such as ‘‘Paper-Author-Paper (PAP)’’ and ‘‘Paper-Subject-Paper (PSP)’’ can link entities within the HIN. Typically, different meta-paths can reveal the varying inter-dependency information between two papers. The meta-path ‘‘PAP’’ indicates that two papers were written by the same author, while ‘‘PSP’’ illustrates that two papers belong to the same subject. Before introducing the proposed model, we summarize the symbols used in Table I in this paper.

III. THE PROPOSED MODEL

In this section, we present overall framework and brief module descriptions, as shown in Figure 3. We then provide detailed explanations for each module following the framework’s order. Finally, we propose different optimization strategies based on the downstream tasks.

A. Network Schema View

This section takes the entire heterogeneous information network as input, as shown in Figure 3 (a). First, we consider each node in the network as an anchor, forming a subnetwork that encompasses all the one-hop neighbor nodes. For example, in Figure 3 (b), the subgraph indicated by the circles represents the subnetwork formed by the target node. Next, we input such subnetworks into a heterogeneous neighbor aggregation network, utilizing attention mechanisms to obtain node embeddings. Finally, we select the embeddings of the desired target node for downstream tasks. **Specific implementation**

process: We project all nodes into the same node space (P_r is projection matrix) due to the various node types within the subnetworks. We define e as the current node computing the attention score, and h_e^0 as its initial embedding. Next, we learn aggregation functions for different types of edges to compute attention scores:

$$\alpha_{ew} = \frac{\exp(f_r(h_e^0, P_r h_w^0))}{\sum_{j \in \mathcal{N}_1(e)} \exp(f_r(h_e^0, P_r h_j^0))}, \quad \forall w \in \mathcal{N}_1(e) \quad (1)$$

where $f_r(\cdot, \cdot)$ denotes the deep neural network designed to implement attention mechanisms over one-hop neighbors. Here, α_{ew} is the influence level for neighbor node w , and $\mathcal{N}_1(e)$ denotes the set of one-hop neighbor nodes of e . Finally, we aggregate the following information from $\mathcal{N}_1(e)$:

$$h_e^1 = \sigma \left(\sum_{w \in \mathcal{N}_1(e)} \alpha_{ew} h_w^0 \right) \quad (2)$$

where σ denotes the activation function.

B. Meta-path View with Generative Learning

In this section, the model inputs are multi-views constructed from different meta-paths. To address the issue of imbalanced data augmentation, we enhance multiple views using a generative mask autoencoder. For example, in Figure 3 (a), we connect two nodes of the same type via the specified meta-path ‘‘PAP’’. Thus, such a graph composed of nodes of the same type is termed a view based on the meta-path ‘‘PAP’’. Based on the preliminaries, we define a set of meta-paths as $\{\rho_1, \dots, \rho_j, \dots, \rho_M\}$, and the corresponding views as $\{G^{\rho_1}, G^{\rho_2}, \dots, G^{\rho_M}\}$. M denotes the number of meta-paths we have selected. Here, the model defines a graph $G = (\mathcal{V}, \mathcal{A}, \mathcal{X})$. For each graph, \mathcal{V} is the node set, $N =$

$|\mathcal{V}|$ is the number of nodes, $\mathcal{A} \in \{0, 1\}^{N \times N}$ is the adjacency matrix, and $\mathbf{X} \in \mathbb{R}^{N \times d}$ is the input node feature matrix.

First, we employ a sampling strategy without replacement to select a set of nodes $\mathcal{V}^{[Mask]}$ from the entire nodes set \mathcal{V} :

$$\mathcal{V}^{[Mask]} = \{\mathcal{V}_i \in \mathcal{V} \mid r_i \leq p\}, r_i \sim \text{Uniform}(0, 1) \quad (3)$$

where p is the mask ratio we set, and \mathcal{V}_i is the node set based on r_i . The uniform random sampling [28] helps the encoder avoid potential bias centers, which refers to the bias arising when all neighbors of a node are either completely masked or completely visible. We mask the entire node features in the set $\mathcal{V}^{[Mask]}$ to obtain $\widetilde{\mathbf{X}}$.

Then, we input the masked features into a GNN encoder to obtain the output. To better reconstruct the features, we set the encoder with adjustable layers L . A multi-layer encoder typically captures distant neighbors to expand their receptive field, enriching their representations.

$$\mathbf{H} = f_E^{(L)}(\mathcal{A}, \widetilde{\mathbf{X}}) \quad (4)$$

where $\widetilde{\mathbf{X}}$ is feature input with the $[Mask]$, and $\mathbf{H} \in \mathbb{R}^{N \times d}$ denotes code after the encoder. Next, to further encourage the decoder to learn compressed representations, we use a new ratio and the same sampling strategy to mask a different set of nodes $\mathcal{V}^{[ReMask]}$. We use a GNN with the same structure as the encoder to act as the decoder. However, the decoder can be GAT [29], GCN [30], and its layer number and type can be different.

$$\mathbf{H}_{Re} = f_D^{(L)}(\mathcal{A}, \widetilde{\mathbf{H}}) \quad (5)$$

where $\widetilde{\mathbf{H}}$ is decoder input with the $[ReMask]$, and \mathbf{H}_{Re} as the reconstructed output under meta-path view. We perform multiple masking operations between encoders, equivalent to the masked nodes reconstructing their features through their neighbors. The uniform sampling strategy ensures that each reconstructed node receives the same supervised signals. Such an autoencoder helps train robust node embeddings and mitigates the impact of noise.

Finally, based on the mentioned meta-path-based multi-view, we denote the reconstructed features obtained from the different views as $\{\mathbf{H}_{Re}^{\rho_1}, \dots, \mathbf{H}_{Re}^{\rho_j}, \dots, \mathbf{H}_{Re}^{\rho_M}\}$. We employ a semantic attention mechanism to assign weights to each view to balance their importance in downstream tasks.

$$w_{\rho_j} = \frac{1}{|\mathcal{V}|} \sum_{e \in \mathcal{V}} \mathbf{q}^T \cdot \tanh (\mathbf{W} \mathbf{h}_{Re}^{\rho_j} + b), \quad (6)$$

where $\mathbf{h}_{Re}^{\rho_j}$ is a reconstructed feature of a node under meta-path ρ_j . \mathbf{q}^T and \mathbf{W} are query vectors and weight matrices to ensure that the dimensions and feature spaces match different views, respectively. Next, the weight for ρ_j , ($j = 1 : N$) is defined as:

$$\gamma_{\rho_j} = \frac{\exp(w_{\rho_j})}{\sum_{j=1}^N \exp(w_{\rho_j})}, \quad (7)$$

where γ_{ρ_j} denotes the weight of meta-path ρ_j . According to the attention mechanism [2], when the semantic attention score γ_{ρ_j} is higher, it indicates the meta-path ρ_j gets more importance in downstream tasks. We aggregate feature embeddings

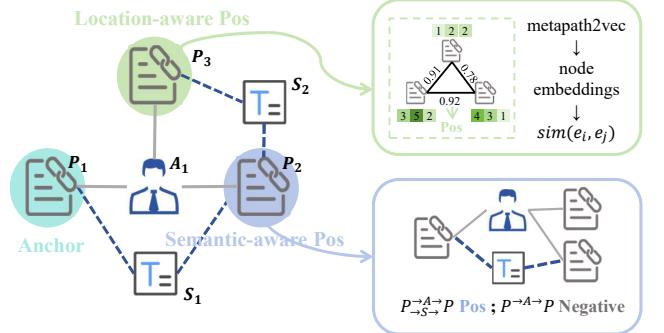


Fig. 4: Our GC-HGNN proposed a hard sampling strategy for inter-contrast. Location-aware Pos: the green node uses metapath2vec [3] to perceive the location within a heterogeneous graph to obtain node embeddings. Semantic-aware Pos: The blue node is connected to the anchor node through all types of meta-paths, indicating the semantic similarity between them.

from all meta-paths to obtain the final node features under the meta-path view.

$$\mathbf{h}_e^2 = \sum_{j=1}^N \gamma_{\rho_j} \cdot \mathbf{h}_e^{\rho_j} \quad (8)$$

The key idea of generative learning lies in feature reconstruction. Therefore, we fit the reconstructed masked node features \mathbf{h}_e^2 and initial features \mathbf{h}_e^0 , using the Mean Squared Error (MSE) [43] as the loss function:

$$\mathcal{L}_{Gen} = \frac{1}{|\mathcal{V}^{[Mask]}|} \sum_{e \in \mathcal{V}^{[Mask]}} (\mathbf{h}_e^0 - \mathbf{h}_e^2)^2 \quad (9)$$

where e denotes the node in the masked set. Previous GAEs [44] typically used a single-layer MLP or Laplacian matrix in decoding and focused more on restoring the graph structure.

C. Hierarchical Contrastive Learning

The inputs for this section are the embeddings derived from the network schema and meta-path, denoted as $\mathbf{h}e^1$ and $\mathbf{h}e^2$. To mitigate sampling bias and information loss within the contrastive discriminator, we introduce innovative sampling strategies and a hierarchical contrast framework. The model utilizes two distinct sampling strategies: node-level contrast for intra-contrast and edge-level contrast for inter-contrast. The process of edge-level sampling is depicted in Figure 4. Subsequently, we obtain an edge sampling set, denoted as \mathcal{E} .

To obtain high-quality positive samples, we propose two sampling strategies: **Location-aware Pos** and **Semantic-aware Pos**, which automatically extract positive samples from the data. As shown in Figure 4, the green nodes represent Location-aware Pos, while the blue nodes denote Semantic-aware Pos.

- Location-aware Pos calculates the similarity between “ $P_1 - P_3$ ” using node embeddings pre-trained by Metapath2vec [3] and selects the top-k nodes to complete the sampling process. Metapath2vec employs meta-path-based random walks to effectively capture structural information in HGs.
- Semantic-aware Pos connects “ $P_1 - P_2$ ” through two meta-paths, “ $P_1 - S_1 - P_2$ ” (PSP) and “ $P_1 - A_1 - P_2$ ” (PAP), rather

than a single meta-path. The more meta-path connections exist, the higher the semantic similarity between them.

We chose metapath2vec for position-aware positives as it captures structural information through meta-path-based walks, making it better suited for long-sequence link awareness. Relying solely on model-based embeddings would weaken position awareness and lose distant node relations [23].

Intra-contrast. First, we utilize sampling set \mathcal{E} to select the positive sample set $\tilde{\mathcal{V}}$ for node-level contrast based on the degree of nodes. High-degree nodes have more connections in the graph and are more likely to capture the central structure and important patterns in the graph. The embeddings of the model, based on multi-views, are denoted by $\{\mathbf{H}_{Re}^{\rho_1}, \dots, \mathbf{H}_{Re}^{\rho_j}, \dots, \mathbf{H}_{Re}^{\rho_M}\}$. Next, we take these embeddings as input to obtain a multi-view loss. Intra-contrast employs Scaled Cosine Error (SCE) [10] as the contrastive loss between multi-views. SCE has been proven to focus on hard samples in the data to improve the stability of representation learning.

$$\mathcal{L}_{Intra} = \frac{1}{|\tilde{\mathcal{V}}|} \sum_{v_i \in \tilde{\mathcal{V}}} \left(1 - \frac{\mathbf{x}_i^T \mathbf{z}_i}{\|\mathbf{x}_i\| \cdot \|\mathbf{z}_i\|} \right)^{\eta}, \eta \geq 1 \quad (10)$$

where \mathbf{x} and \mathbf{z} respectively denote embeddings of the same node under two meta-path views. During training, we can scale the cosine error by a power of $\eta \geq 1$ to minimize the differences of important samples under multi-views.

Inter-contrast. Inter-contrast performs collaborative contrastive learning between two non-aligned views. In set \mathcal{E} , we define the target node to be sampled as i and the connected nodes to it as a positive set \mathbb{P}_i , inspired by [7]. Otherwise, they form the negative sample set \mathbb{N}_i . From the scales of \mathcal{E} and $\tilde{\mathcal{V}}$, inter-contrast typically has more hard negative samples, and thus, its supervisory role is more evident. The contrastive loss of the network schema view with respect to the meta-path view is as follows:

$$\mathcal{L}_{Inter}^1 = -\log \frac{\sum_{j \in \mathbb{P}_i} \exp(\text{sim}(h_i^1, h_j^2) / \tau)}{\sum_{k \in \{\mathbb{P}_i \cup \mathbb{N}_i\}} \exp(\text{sim}(h_i^1, h_k^2) / \tau)} \quad (11)$$

where $\text{sim}(i, j)$ is the cosine similarity, and τ denotes a temperature parameter.

Inter-contrast utilizes mutual contrast between the two views to achieve the collaborative process. The contrastive loss of the meta-path view with respect to the network schema view is \mathcal{L}_{Inter}^2 . The overall inter-contrast losses are as follows:

$$\mathcal{L}_{Inter} = \frac{1}{|\mathcal{E}|} \sum_{i \in \mathcal{E}} [\lambda \cdot \mathcal{L}_{Inter}^1 + (1 - \lambda) \cdot \mathcal{L}_{Inter}^2] \quad (12)$$

where λ is a balance coefficient to control the weight of each view pair. We employ intra-contrast and inter-contrast for hierarchical contrastive learning and analyze their operational mechanisms. The intra-contrast assumes that meta-paths are interconnected and aims to maximize their mutual information [14]. This method integrates the different semantics across meta-paths to form a generalized knowledge representation for downstream tasks. On the other hand, the inter-contrast

approach attempts to enhance node representation by contrasting local and global information, preventing nodes from being influenced by noise from higher-order neighbors. The hierarchical contrast and sampling strategies are designed to construct a contrastive discriminator to enhance the adversarial interplay between the generative and contrastive models.

D. Model Optimization

In this subsection, we describe the optimization strategies for different tasks in our experiments. It has two aspects: node classification and link prediction.

Node classification. First, the common practice for the node classification task [7], [45] is to train embeddings for all nodes using self-supervised loss. Then, we fine-tune the model on different datasets using a few labeled samples and a single-layer MLP. The node-level and edge-level strategies we propose typically lead to considerable loss differences. Therefore, it is necessary to balance their weights during the training phase. The loss generated by hierarchical contrastive learning is as follows:

$$\mathcal{L}_{hcl} = \lambda_{Intra} \cdot \mathcal{L}_{Intra} + \lambda_{Inter} \cdot \mathcal{L}_{Inter} + \lambda_{Gen} \cdot \mathcal{L}_{Gen} \quad (13)$$

where λ_{Intra} , λ_{Inter} and λ_{Gen} are hyper-parameters to balance the influences of losses.

Link prediction. Next, for the link prediction task [4], [5], we introduce the Bayesian Personalized Ranking Loss (BPR) [46]. This loss function treats two connected nodes as a positive pair and two unconnected nodes as a negative pair. Its objective is to maximize the scores between positive pairs and negative pairs. Formally, the BPR loss is defined as follows:

$$\mathcal{L}_{bpr} = -\frac{1}{|\mathcal{O}|} \sum_{(i,j,k) \in \mathcal{O}} \log \sigma(\mathbf{d}_i^\top \mathbf{d}_j - \mathbf{d}_i^\top \mathbf{d}_k) \quad (14)$$

where $\mathcal{O} = \{(i, j, k) \mid A_{i,j} = 1, A_{i,k} = 0\}$ is the training data. We combine \mathcal{L}_{hcl} and \mathcal{L}_{bpr} to get the loss:

$$\mathcal{L} = \lambda_{hcl} \cdot \mathcal{L}_{hcl} + \lambda_{bpr} \cdot \mathcal{L}_{bpr} \quad (15)$$

where λ_{hcl} and λ_{bpr} denote hyperparameters used to balance task weights.

E. Time Complexity Analysis

The network schema view's time complexity is $O(|V| \cdot D_n d^2)$, where $|V|$ denotes the edge number of a heterogeneous graph. D_n is the average node degree and d is the feature dimension. For the meta-path view, GC-HGNN has M meta-path-based subgraphs. So, the time complexity of the meta-path view is $O(|\mathcal{V}| \cdot 2MLDd)$, where L is the number of GNN layers. Then, the self-supervised loss contains λ_{Intra} and λ_{Inter} . The time complexity to get positive samples and calculate λ_{Intra} is $O(|\tilde{\mathcal{V}}| \cdot d)$. For the λ_{Inter} , the complexity is $O(|\mathcal{V}|^2 \cdot d)$. Because the edge set \mathcal{E} is a matrix $\mathcal{E} \in \{0, 1\}^{N \times N}$ composed of target nodes. In the link prediction task, the complexity of the additional \mathcal{L}_{bpr} is $O(|\mathcal{O}| \cdot d)$. Thus, the overall time complexity of GC-HGNN is $O((|V| \cdot D_n d + |\mathcal{V}| \cdot 2MLD_m + |\tilde{\mathcal{V}}| + |\mathcal{V}|^2 + |\mathcal{O}|) \cdot d)$. This complexity is the time complexity for each epoch during

TABLE II: Time and resource comparison on datasets.

Dataset	ACM	DBLP	Aminer	Freebase
Time (epoch/s)				
HeCo [7]	0.204	0.198	0.361	0.272
HGMAE [23]	0.212	0.334	0.203	0.114
GC-HGNN	0.033	0.058	0.043	0.037
Resource (memory/G)				
HeCo [7]	2.21	3.25	8.75	6.36
HGMAE [23]	5.49	6.82	14.48	8.93
GC-HGNN	2.54	4.10	8.33	6.65

TABLE III: Statistics of datasets used in this paper.

Dataset	Nodes	Edges	Relations	Task
ACM	11,246	34,852	2	Node Classification
DBLP	26,128	239,566	3	Node Classification
Aminer	55,783	153,676	2	Node Classification
Freebase	43,854	151,043	3	Node Classification
Last.fm	31,469	913,216	3	Link Prediction
Yelp	31,092	976,276	5	Link Prediction
Douban Movie	37,595	3,429,882	6	Link Prediction
Douban Book	51,029	4,451,894	7	Link Prediction

training, and the number of training epochs will determine the overall training time. We compared several strong baselines in terms of training time. Our model performs faster per epoch due to optimized modules, demonstrating lower complexity and greater scalability compared to baseline methods.

IV. EXPERIMENT

In this section, we conduct extensive experiments and answer the following research questions:

- **RQ1:** How does GC-HGNN perform w.r.t. tasks of node classification and link prediction?
- **RQ2:** Are the key components in our GC-HGNN delivering the expected performance gains?
- **RQ3:** How does GC-HGNN combine contrast and generation to achieve performance beyond one or the other?
- **RQ4:** How do different settings influence the effectiveness of GC-HGNN?

A. Experimental Settings

We followed the experimental benchmarks of HeCo and HGMAE, where the hidden layer dimension is fixed at 128, and the number of fine-tuning samples per class is set to 20, 40, and 60. The original HGMAE paper used a hidden layer dimension of 512, for which we readjusted the optimal parameters. Due to space limitations, we reported the results for 20 samples, as labels are usually valuable, and fewer samples often require stronger model generalization ability. All experimental results were repeated 10 times on the 4090, and we release the reproducible framework for the model as open source at the code¹.

¹<https://github.com/wangyu0627/HGNN-baselines>

B. Datasets

The performance of GC-HGNN has been validated on six real-world datasets. These datasets encompass four datasets for node classification: ACM, DBLP, Aminer, and Freebase, as well as four datasets for link prediction: Last.fm, Yelp, Douban Movie and Douban Book. A statistical overview of all the datasets is presented in Table III.

Datasets for Node Classification.

- **ACM** [47]. The target nodes are papers (3 classes), averagely linked to 3.33 authors and one subject.
- **DBLP** [33]. The target nodes are authors (4 classes), averagely linked to 4.84 papers.
- **Freebase** [23]. The target nodes are movies (3 classes), which are typically linked to an average of 18.7 actors, 1.07 directors, and 1.83 writers.
- **Aminer** [7]. The target nodes are papers (4 classes), averagely linked to 2.74 authors and 8.96 references.

Datasets for Link Prediction.

- **Last.fm**² predicts user-artist interactions in a music recommendation setting, with other node type including tag.
- **Yelp**³ predicts user-business interactions in a business recommendation setting, with node types including category, city and category,etc.
- **Douban Movie**⁴ predicts user-movie interactions in a movie recommendation setting, with node types including actor, director and type, etc.
- **Douban Book** [4] predicts user-book interactions in a book recommendation setting, with node types including author, publisher and group, etc.

C. Baselines

We respectively select the most representative baseline model for node classification and link prediction.

Heterogeneous Graph Representation Learning Models.

- **DeepWalk** [48] maps network structure to vector space via random walks, preserving neighbor similarity.
- **MP2vec** [3] employs meta-path-guided random walks for heterogeneous network embeddings.
- **HERec** [4] integrates meta-path-based random walks for node embeddings in HINs with fusion functions.
- **HAN** [2] employs node-level and semantic-level attention to capture node and meta-path importance.
- **HGT** [36] introduces a novel architecture for handling heterogeneous structures in Web-scale graphs.
- **DMGI** [45] achieves self-supervised learning by meta-path-based networks and mutual information between networks.
- **HeCo** [7] uses collaborative supervised contrast between network schema view and meta-path view.
- **MHGNC** [49] utilizes multi-layer convolution aggregation to capture meta-path interactions of different lengths.
- **HGCM** [17] uses meta-paths to generate multi-views combined with comparative learning between views.

²<https://www.last.fm/>

³<https://www.yelp.com/>

⁴<https://m.douban.com/>

TABLE IV: Quantitative results (%) on node classification. In the table, the symbols \mathbf{X} , \mathbf{A} , and \mathbf{Y} denote the node features, graph structure, and predicted node labels, respectively.

Methods	Data	ACM		DBLP		Aminer		Freebase	
		Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
DeepWalk(2017)	\mathbf{A}	68.34 \pm 0.42	74.05 \pm 0.47	89.24 \pm 0.49	90.12 \pm 0.45	68.48 \pm 0.38	74.00 \pm 0.41	59.20 \pm 0.50	59.85 \pm 0.43
MP2vec(2017)	\mathbf{A}	68.05 \pm 0.98	72.86 \pm 0.92	89.98 \pm 0.84	90.30 \pm 0.38	66.28 \pm 1.17	72.21 \pm 0.84	55.50 \pm 0.49	53.17 \pm 0.34
HERec(2018)	\mathbf{A}	68.84 \pm 0.44	74.45 \pm 0.45	89.44 \pm 0.67	90.37 \pm 0.47	68.73 \pm 0.83	74.21 \pm 0.87	59.46 \pm 1.16	60.11 \pm 0.46
HAN(2019)	$\mathbf{X}, \mathbf{A}, \mathbf{Y}$	88.30 \pm 0.23	88.20 \pm 0.37	86.00 \pm 0.53	85.18 \pm 0.49	64.15 \pm 0.35	75.60 \pm 0.79	58.50 \pm 0.93	61.30 \pm 0.41
HGT(2020)	$\mathbf{X}, \mathbf{A}, \mathbf{Y}$	87.09 \pm 0.35	88.20 \pm 0.26	91.36 \pm 0.23	91.24 \pm 0.49	64.15 \pm 0.35	75.60 \pm 0.24	58.30 \pm 0.72	61.34 \pm 0.18
MHGNC(2022)	$\mathbf{X}, \mathbf{A}, \mathbf{Y}$	88.86 \pm 0.63	88.70 \pm 0.22	90.62 \pm 0.58	90.62 \pm 0.19	71.34 \pm 0.10	79.24 \pm 0.79	57.12 \pm 0.32	59.30 \pm 0.33
DMGI(2020)	\mathbf{X}, \mathbf{A}	87.66 \pm 0.24	87.60 \pm 0.35	89.55 \pm 0.52	90.66 \pm 0.46	63.15 \pm 0.21	66.61 \pm 0.38	55.99 \pm 0.59	59.37 \pm 0.27
HeCo(2021)	\mathbf{X}, \mathbf{A}	88.56 \pm 0.28	88.71 \pm 0.43	90.26 \pm 0.37	90.59 \pm 0.25	71.64 \pm 0.22	79.95 \pm 0.55	59.87 \pm 0.33	62.33 \pm 0.40
HGML(2023)	\mathbf{X}, \mathbf{A}	88.46 \pm 0.38	87.90 \pm 0.27	90.47 \pm 0.33	91.11 \pm 0.49	71.84 \pm 0.45	78.30 \pm 0.49	58.92 \pm 2.57	62.71 \pm 0.23
HGMAE(2023)	\mathbf{X}, \mathbf{A}	89.29 \pm 0.36	89.15 \pm 0.31	91.60 \pm 0.58	91.89 \pm 0.35	72.63 \pm 0.27	79.87 \pm 0.42	60.82 \pm 0.51	63.89 \pm 0.38
GC-HGNN	\mathbf{X}, \mathbf{A}	90.54 \pm 0.28	90.36 \pm 0.31	92.54 \pm 0.36	93.01 \pm 0.30	73.44 \pm 0.20	81.28 \pm 0.32	62.47 \pm 0.42	65.85 \pm 0.35

TABLE V: Quantitative results on link prediction.

Dataset	Douban-Book		Douban-Movie		Last.fm		Yelp	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
BPRMF(2012)	0.1302	0.1112	0.1678	0.1888	0.2400	0.2349	0.0752	0.0466
HERec(2018)	0.1313	0.1201	0.1714	0.1919	0.2526	0.2484	0.0822	0.0471
LightGCN (2020)	0.1412	0.1307	0.1791	0.2011	0.2623	0.2638	0.0893	0.0565
SGL-ED (2021)	0.1685	<u>0.1631</u>	0.1890	0.2056	0.2724	0.2704	0.0971	0.0614
NCL (2022)	0.1694	0.1623	0.1910	0.2074	0.2738	0.2718	0.0975	<u>0.0623</u>
VGCL (2023)	<u>0.1701</u>	0.1602	<u>0.1933</u>	<u>0.2097</u>	0.2740	<u>0.2726</u>	<u>0.0978</u>	0.0621
HAN (2019)	0.1290	0.1137	0.1676	0.1884	0.2443	0.2397	0.0767	0.0449
HGT (2020)	0.1308	0.1157	0.1696	0.1904	0.2463	0.2417	0.0787	0.0469
SMIN (2021)	0.1353	0.1239	0.1718	0.1909	0.2547	0.2560	0.0847	0.0485
HGCL (2023)	0.1526	0.1426	0.1868	0.2024	0.2684	0.2688	0.0954	0.0607
GC-HGNN	0.1890	0.1717	0.1997	0.2175	0.2813	0.2795	0.1044	0.0649
Improv.	11.11%	7.18%	3.31%	3.72%	2.63%	2.53%	6.75%	4.51%

- **HGMAE** [23] captures graph information by reconstructing masked meta-path-based view and embedding.

Graph Recommendation Models.

- **BPRMF** [46] learns implicit feature vectors between users and items to maximize difference in pairwise preferences.
- **LightGCN** [30] removes the feature transformation and nonlinear activations of GCN to achieve recommendation.
- **SGL** [15] generates contrast views by edge dropout to aid contrastive learning to enhance recommendation.
- **SMIN** [6] utilizes metagraph informax network to enhance user preference representation for social recommendation.
- **NCL** [20] considers the relationship between neighbor nodes to enhance collaborative filtering.
- **HGCL** [5] integrates heterogeneous relational semantics by leveraging contrastive self-supervised learning.
- **VGCL** [16] introduces a variational graph reconstruction for graph augmentation to achieve robust recommendation.

D. Overall Performance (RQ1)

Node Classification. The experiment results are shown in Table IV. Notably, self-supervised methods generally outperform unsupervised ones based on shallow networks. This performance disparity arises because unsupervised models only consider the graph structure, whereas self-supervised models endeavor to mine hidden semantic relationships from the HIN. Conversely, the performance of supervised models is dependent on more labels, which goes against the challenge

of label scarcity and preciousness. HGMAE, a generative reconstruction model, demonstrates the significant potential of autoencoders in graph representation learning, while HeCo, a contrastive model, also yields remarkable results. These two models learn graph representations from different perspectives, providing insights for designing future GNN models. Inspired by these insights, GC-HGNN integrates the generative masked autoencoder with a contrastive discriminator to improve performance by generating more hard negative samples.

Link Prediction. Recommendation is a classic task for heterogeneous link prediction. Considering the task adaptability, we use LightGCN [30] to replace the network schema. Here, we standardize the embedding dimension to 64 and select the recall and NDCG [50] as evaluation metrics. The results are shown in Table V. It can be observed that NCL and HGCL, as representatives of contrastive learning, demonstrate distinct advantages over other models within their respective categories. The essence of contrastive learning lies in achieving uniformity and alignment. However, modifying the graph structure to achieve uniformity is time-consuming and tricky, so it is better to consider this aspect from the perspective of the embedding space. Our GC-HGNN incorporates embeddings learned from the meta-path view into the network schema view and performs hierarchical contrastive learning. Finally, across four datasets, our model achieved improvements of over 3% compared to the strongest baseline models.

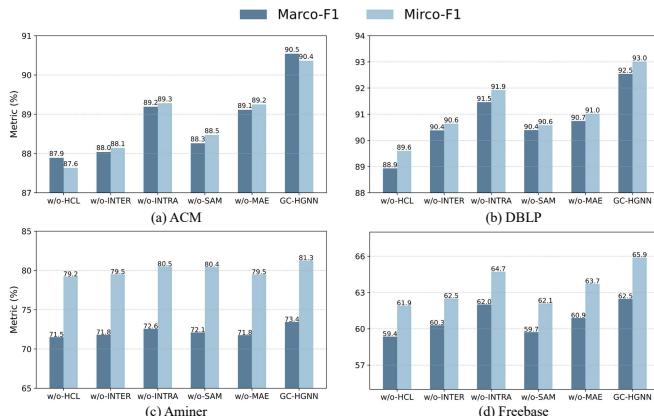


Fig. 5: The comparison of GC-HGNN and its variants.

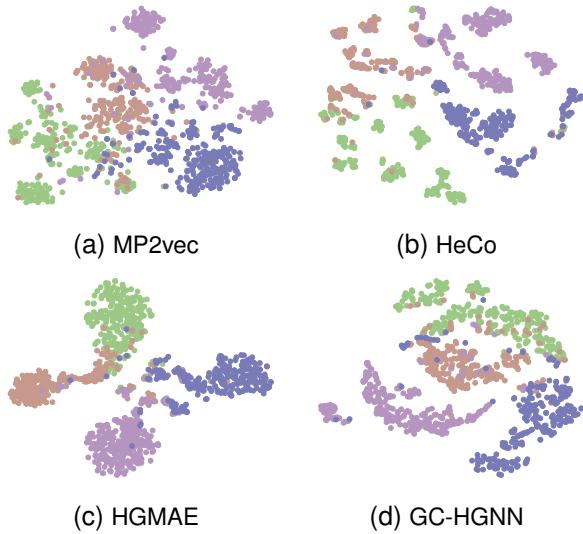


Fig. 6: Embedding visualization of DBLP dataset. Different colors indicate different node category labels.

E. Ablation Study(RQ2)

In this section, we conducted ablation experiments to validate the effectiveness of key components in our GC-HGNN and provided possible explanations for the experimental results. We present three variants of our model.

- **w/o-MAE:** This variant replaces the meta-path view's masked autoencoder (MAE) with a normal GAT [29] encoder. Because usually GAT has stronger performance than GCN [30] on node classification tasks.
- **w/o-SAM:** This variant replaces the sampling strategy (SAM) of GC-HGNN with random sampling while maintaining the same sampling ratio.
- **w/o-INTRA:** This variant replaces the intra-contrast corresponds to Eq. 10.
- **w/o-INTER:** This variant cancels the inter-contrast integrated from Eqs. 11 and 12.
- **w/o-HCL:** The last cancels the hierarchical contrastive learning (HCL) module and directly uses the InfoNCE [39] loss for downstream tasks.

As shown in Figure 5, in all cases, removing the HCL module results in poorest performance compared to other

variants. Additionally, variants where SAM or HCL is removed generally exhibit the worse performance in most cases. This suggests that inter-contrast through aligned local and global heterogeneous information significantly enhances semantics, while intra-contrast regulates semantic uniformity. We further highlights two important points: 1) The graph structural information is overemphasized, and reconstructing the graph links is not necessary; 2) The existing heterogeneous contrastive learning strategies are insufficient to serve as effective contrastive discriminators to enhance the masked autoencoder. It is noteworthy that 'INTER' serves as the principal loss function. This suggests that aligning high-order and first-order semantic representations of heterogeneous information can significantly improve the prediction accuracy, while contrast between meta-paths regulates semantic consistency [41].

F. Interpretability and Visualization (RQ3)

In this study, we examine the various modules of GC-HGNN, attempting to provide a rational explanation. We investigated the advantages of the generative-contrastive model and then further illustrated the motivation through the representation visualization.

1) *Interpretability:* Generative-contrastive (GC) representation learning [16], [18], [19] employs discriminative loss functions as its objectives, aiming to address some inherent shortcomings of generative reconstruction, such as sensitive and conservative distribution issues and low-level abstraction objectives. In contrastive learning [7], [9], [15], [39], we can alleviate these issues by setting up appropriate contrastive discriminators to learn distinguishable information for different samples. However, the encoder and decoder in generative learning [11], [12], [23], [28] provide powerful embedding expression capabilities. The decoder requires the representation to be reconstructable, meaning that it contains all the information of the input. Furthermore, GC-based self-supervised methods eschew the point-to-point reconstruction objective and aim to align distributions in the data. In summary, These methods [14] integrate the advantages of both generative and contrastive methods, yet confront the challenge of constructing a framework within the field of heterogeneous graphs. We propose a masked autoencoder to enhance the expressive power of meta-path perspectives, followed by the use of hierarchical contrastive learning and sampling strategies to serve as enhanced contrastive discriminators. Our experiments demonstrate that this is an effective and novel exploration.

2) *Visualization:* To gain a more intuitive understanding of the embeddings from different models, we employ t-SNE [51] to reduce the dimensionality of the embeddings and visualize them with labels. As illustrated in Figure 6, we select three representative models: unsupervised MP2vec [3], contrastive HeCo [7], and generative HGMAE [23]. We can observe that MP2vec performs the worst, with different types of nodes mixed together, making them difficult to distinguish. HeCo effectively differentiate categories, but each block remains separated and does not form distinct clusters. HGMAE uses a hidden dimension of 256 and is a generative method, thus exhibiting more evident clustering phenomena. However, our

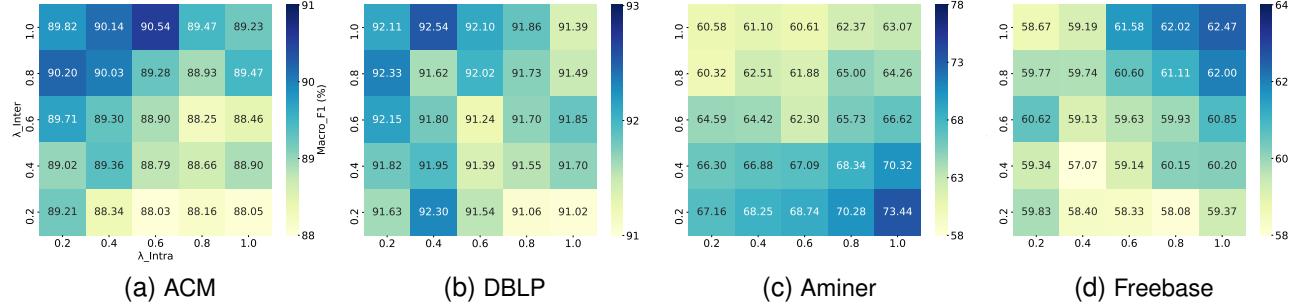


Fig. 7: The impact wrt. hierarchical contrast coefficient. The axes on the left belong to the ACM and DBLP datasets, and the axes on the right belong to other datasets.

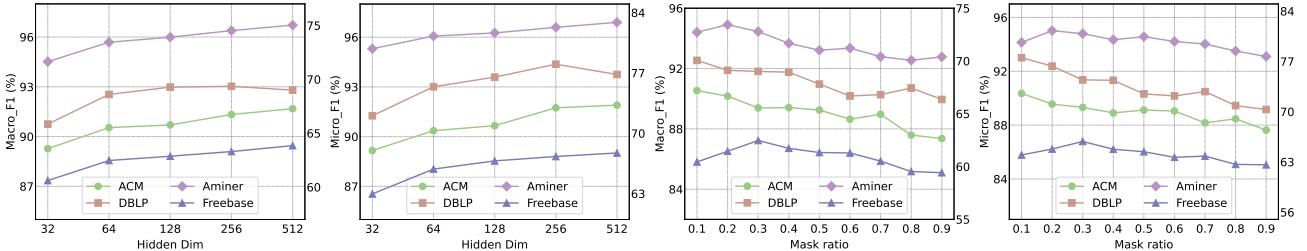


Fig. 8: The impact wrt. dimension and mask ratio. The axes on the left belong to the ACM and DBLP datasets, and the axes on the right belong to other datasets.

TABLE VI: Performance comparison with coefficient \mathcal{L}_{Gen} .

Dataset	ACM		DBLP		Aminer		Freebase	
	Lambda	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1
0.1	90.54±0.28	90.36±0.31	92.54±0.36	93.01±0.30	72.44±0.30	80.28±0.34	61.47±0.37	64.46±0.43
0.2	89.50±0.32	89.48±0.29	91.37±0.41	92.24±0.34	73.25±0.20	81.45±0.32	62.40±0.42	65.85±0.35
0.3	88.47±0.33	87.72±0.29	90.23±0.40	90.92±0.34	71.34±0.18	79.16±0.26	60.62±0.47	63.74±0.41
0.4	86.66±0.29	86.20±0.32	87.33±0.37	87.29±0.31	66.21±0.21	73.35±0.31	57.47±0.43	60.64±0.36
0.5	80.51±0.32	80.25±0.38	82.43±0.29	82.98±0.33	63.41±0.25	71.18±0.27	55.38±0.30	59.14±0.47

TABLE VII: Performance comparison of different encoder and decoder variants.

Encoder	Decoder	ACM		DBLP		Aminer		Freebase	
		Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
GAT	GAT	90.29 ± 0.22	90.16 ± 0.30	92.14 ± 0.32	92.49 ± 0.41	73.44 ± 0.20	81.28 ± 0.32	62.47 ± 0.42	65.85 ± 0.35
GCN	GCN	90.54 ± 0.28	90.36 ± 0.31	91.16 ± 0.24	91.52 ± 0.27	71.90 ± 0.32	79.48 ± 0.43	61.85 ± 0.20	64.88 ± 0.24
GAT	GCN	89.24 ± 0.27	89.46 ± 0.32	90.74 ± 0.18	91.15 ± 0.23	72.69 ± 0.25	80.44 ± 0.34	61.23 ± 0.33	63.78 ± 0.26
GCN	GAT	89.68 ± 0.18	89.61 ± 0.36	92.54 ± 0.36	93.01 ± 0.30	72.31 ± 0.44	79.98 ± 0.21	62.02 ± 0.29	65.24 ± 0.35

GC-HGNN, based on a contrastive framework, demonstrates both clustering and uniform distribution, effectively balancing the advantages of contrastive and generative approaches. This phenomenon further indicates the effectiveness of our GC-based paradigm.

G. Parameter Sensitivity (RQ4)

In this study, we explore the impact of key hyperparameters on heterogeneous graph node classification. Moreover, we investigate the effects of hierarchical contrast coefficient, embedding dimension, and mask ratio.

- Hierarchical contrast coefficient.** The \mathcal{L}_{hcl} contains λ_{Intra} and λ_{Inter} , which are set from 0 to 1 with an incremental step. In Figure 7, it is observed that the performance improves as λ_{Inter} increases. This phenomenon aligns with

the principle of our proposed sampling method, where inter-contrast has more hard negative samples.

- Generative coefficient.** In Table VI, we evaluated \mathcal{L}_{Gen} ranging from 0.1 to 0.5, observing that performance metrics typically decline as the \mathcal{L}_{Gen} increases. It can be observed that the model tends to favor a smaller (generative learning) GL loss, which aligns with our idea regarding the auxiliary role of GL in the (generative-contrastive) GC paradigm.

- Embedding dimension.** We examine how different embedding sizes affect node classification performance ranging from 32 to 512. Figure 8 shows that most datasets show improvement up to a 512-dimensional embedding, apart from DBLP, which peaks at 256. However, to balance the marginal gains against higher computational costs and overfitting risks, we standardize the embedding size at 64

for all tests.

- **Mask ratio.** We utilize a masked autoencoder, where the mask ratio plays a crucial role as a hyper-parameter in generating negative samples. As shown in Figure 8, our model achieves better results with a lower mask ratio, contrary to the conclusions drawn by HGMAE [23]. We attribute this to the fact that, compared to generative feature reconstruction, the game between contrastive learning and generation learns to reconstruct the original data distribution by minimizing the distributional divergence rather than reconstructing individual samples [14].
- **Encoder types in MAE.** We set multiple variants for the encoder and decoder to test their effect on GC-HGNN. Theoretically, our encoder can be any GNN model (such as GIN [52]), or even an MLP. As shown in Table VII, the best result is highlighted in bold, while the second-best is underlined. We choose a combination of the commonly used GAT [29] and GCN [53]. It can be observed that the GAT-GAT combination achieves the best or second-best performance on most datasets. This phenomenon aligns with the conclusion that the attention mechanism can effectively capture relationships between nodes in node classification tasks. Moreover, all variants outperform the MAE-free variant from the ablation study, demonstrating that MAE effectively mitigates noise limitation.

In general, the findings suggest that higher inter-contrast, higher embedding dimensions, and lower mask ratios typically yield better results, with some dataset-specific variations.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel generative-contrastive heterogeneous graph neural network termed GC-HGNN. It represented a generative learning enhanced contrastive paradigm and a meaningful exploration. We combined metapaths and masked autoencoders to create augmented contrastive views without disrupting the data structure. The generative-contrastive framework significantly improves the performance of node classification and link prediction tasks. We compared our model with seventeen state-of-the-art models on eight real-world datasets, demonstrating that it outperformed the them on these tasks.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation of China (No. 62206002, No. 62272001 and No. 62206004), Hefei Key Common Technology Project (NO. 2023SGJ014), Natural Science Foundation of Anhui Province (No. 2208085QF195) and Xunfei Zhiyuan Digital Transformation Innovation Research Special for Universities (NO. 2023ZY001).

REFERENCES

- [1] Y. Sun and J. Han, “Mining heterogeneous information networks: A structural analysis approach,” *KDD*, vol. 14, no. 2, pp. 20–28, 2013.
- [2] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, “Heterogeneous graph attention network,” in *WWW*, 2019, pp. 2022–2032.
- [3] Y. Dong, N. V. Chawla, and A. Swami, “metapath2vec: Scalable representation learning for heterogeneous networks,” in *KDD*, 2017, pp. 135–144.
- [4] C. Shi, B. Hu, W. X. Zhao, and S. Y. Philip, “Heterogeneous information network embedding for recommendation,” *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 31, no. 2, pp. 357–370, 2018.
- [5] M. Chen, C. Huang, L. Xia, W. Wei, Y. Xu, and R. Luo, “Heterogeneous graph contrastive learning for recommendation,” in *WSDM*, 2023, pp. 544–552.
- [6] X. Long, C. Huang, Y. Xu, H. Xu, P. Dai, L. Xia, and L. Bo, “Social recommendation with self-supervised metagraph informax network,” in *CIKM*, 2021, pp. 1160–1169.
- [7] X. Wang, N. Liu, H. Han, and C. Shi, “Self-supervised heterogeneous graph neural network with co-contrastive learning,” in *KDD*, 2021, pp. 1726–1736.
- [8] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, “Pathsim: Meta path-based top-k similarity search in heterogeneous information networks,” *VLDB*, vol. 4, no. 11, pp. 992–1003, 2011.
- [9] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *CVPR*, 2020, pp. 9729–9738.
- [10] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, “Bootstrap your own latent-a new approach to self-supervised learning,” *NeurIPS*, vol. 33, pp. 21271–21284, 2020.
- [11] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, “Palm: Scaling language modeling with pathways,” *JMLR*, vol. 24, no. 240, pp. 1–113, 2023.
- [12] J. D. M.-W. C. Kenton and L. K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL-HLT*, 2019, pp. 4171–4186.
- [13] Y. Zhu, Y. Xu, H. Cui, C. Yang, Q. Liu, and S. Wu, “Structure-enhanced heterogeneous graph contrastive learning,” in *SDM*. SIAM, 2022, pp. 82–90.
- [14] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, “Self-supervised learning: Generative or contrastive,” *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 35, no. 1, pp. 857–876, 2021.
- [15] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, “Self-supervised graph learning for recommendation,” in *SIGIR*, 2021, pp. 726–735.
- [16] Y. Yang, Z. Wu, L. Wu, K. Zhang, R. Hong, Z. Zhang, J. Zhou, and M. Wang, “Generative-contrastive graph learning for recommendation,” in *SIGIR*, 2023, pp. 1117–1126.
- [17] Z. Wang, Q. Li, D. Yu, X. Han, X.-Z. Gao, and S. Shen, “Heterogeneous graph contrastive multi-view learning,” in *SDM*. SIAM, 2023, pp. 136–144.
- [18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *ACM Communications*, vol. 63, no. 11, pp. 139–144, 2020.
- [19] C. Wu, C. Wang, J. Xu, Z. Liu, K. Zheng, X. Wang, Y. Song, and K. Gai, “Graph contrastive learning with generative adversarial network,” in *KDD*, 2023, pp. 2721–2730.
- [20] Z. Lin, C. Tian, Y. Hou, and W. X. Zhao, “Improving graph collaborative filtering with neighborhood-enriched contrastive learning,” in *WWW*, 2022, pp. 2320–2329.
- [21] Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus, “Hard negative mixing for contrastive learning,” *NeurIPS*, vol. 33, pp. 21798–21809, 2020.
- [22] Y. Liu, X. Yang, S. Zhou, X. Liu, Z. Wang, K. Liang, W. Tu, L. Li, J. Duan, and C. Chen, “Hard sample aware network for contrastive deep graph clustering,” in *AAAI*, vol. 37, no. 7, 2023, pp. 8914–8922.
- [23] Y. Tian, K. Dong, C. Zhang, C. Zhang, and N. V. Chawla, “Heterogeneous graph masked autoencoders,” in *AAAI*, vol. 37, no. 8, 2023, pp. 9997–10005.
- [24] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *ICML*. PMLR, 2017, pp. 1263–1272.
- [25] W. Wei, X. Ren, J. Tang, Q. Wang, L. Su, S. Cheng, J. Wang, D. Yin, and C. Huang, “Llmrec: Large language models with graph augmentation for recommendation,” in *WSDM*, 2024, pp. 806–815.
- [26] J. Zhuo, F. Qin, C. Cui, K. Fu, B. Niu, M. Wang, Y. Guo, C. Wang, Z. Wang, X. Cao *et al.*, “Improving graph contrastive learning via adaptive positive sampling,” in *CVPR*, 2024, pp. 23179–23187.

- [27] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, “Graph contrastive learning with augmentations,” *NeurIPS*, vol. 33, pp. 5812–5823, 2020.
- [28] Z. Hou, X. Liu, Y. Cen, Y. Dong, H. Yang, C. Wang, and J. Tang, “Graphmae: Self-supervised masked graph autoencoders,” in *KDD*, 2022, pp. 594–604.
- [29] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *ICLR*, 2018.
- [30] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, “Lightgcn: Simplifying and powering graph convolution network for recommendation,” in *SIGIR*, 2020, pp. 639–648.
- [31] L. Sang, Y. Wang, Y. Zhang, and X. Wu, “Denoising heterogeneous graph pre-training framework for recommendation,” *ACM Transactions on Information Systems (TOIS)*, 2024.
- [32] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, “Heterogeneous graph neural network,” in *KDD*, 2019, pp. 793–803.
- [33] X. Fu, J. Zhang, Z. Meng, and I. King, “Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding,” in *WWW*, 2020, pp. 2331–2341.
- [34] L. Yu, L. Sun, B. Du, C. Liu, W. Lv, and H. Xiong, “Heterogeneous graph representation learning with relation awareness,” *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2022.
- [35] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, “Modeling relational data with graph convolutional networks,” in *ESWC*, 2018, pp. 593–607.
- [36] Z. Hu, Y. Dong, K. Wang, and Y. Sun, “Heterogeneous graph transformer,” in *WWW*, 2020, pp. 2704–2710.
- [37] M. Li, J. Shi, L. Bai, C. Huang, Y. Jiang, K. Lu, S. Wang, and E. R. Hancock, “Frameerc: Framelet transform based multimodal graph neural networks for emotion recognition in conversation,” *Pattern Recognition*, vol. 161, p. 111340, 2025.
- [38] J. Li, R. Zheng, H. Feng, M. Li, and X. Zhuang, “Permutation equivariant graph framelets for heterophilous graph learning,” *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, vol. 35, no. 9, pp. 11 634–11 648, 2024.
- [39] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *ICML*. PMLR, 2020, pp. 1597–1607.
- [40] Y. Ren and B. Liu, “Heterogeneous deep graph infomax,” in *AAAI Workshop*, 2020.
- [41] L. Sang, Y. Wang, Y. Zhang, Y. Zhang, and X. Wu, “Intent-guided heterogeneous graph contrastive learning for recommendation,” *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 37, no. 4, pp. 1915–1929, 2025.
- [42] J. Zheng, Q. Ma, H. Gu, and Z. Zheng, “Multi-view denoising graph auto-encoders on heterogeneous information networks for cold-start recommendation,” in *KDD*, 2021, pp. 2338–2348.
- [43] Z. Hu, Y. Dong, K. Wang, K.-W. Chang, and Y. Sun, “Gpt-gnn: Generative pre-training of graph neural networks,” in *KDD*, 2020, pp. 1857–1867.
- [44] T. N. Kipf and M. Welling, “Variational graph auto-encoders,” *arXiv preprint arXiv:1611.07308*, 2016.
- [45] C. Park, D. Kim, J. Han, and H. Yu, “Unsupervised attributed multiplex network embedding,” in *AAAI*, vol. 34, no. 04, 2020, pp. 5371–5378.
- [46] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “Bpr: Bayesian personalized ranking from implicit feedback,” *arXiv preprint arXiv:1205.2618*, 2012.
- [47] H. Han, T. Zhao, C. Yang, H. Zhang, Y. Liu, X. Wang, and C. Shi, “Openhgnn: An open source toolkit for heterogeneous graph neural network,” in *CIKM*, 2022, pp. 3993–3997.
- [48] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *KDD*, 2014, pp. 701–710.
- [49] P. Yu, C. Fu, Y. Yu, C. Huang, Z. Zhao, and J. Dong, “Multiplex heterogeneous graph convolutional network,” in *KDD*, 2022, pp. 2377–2387.
- [50] Y. Zhang, Y. Zhang, D. Yan, S. Deng, and Y. Yang, “Revisiting graph-based recommender systems from the perspective of variational auto-encoder,” *ACM Transactions on Information Systems (TOIS)*, vol. 41, no. 3, pp. 1–28, 2023.
- [51] L. van der Maaten and G. Hinton, “Visualizing high-dimensional data using t-sne,” *JMLR*, vol. 9, no. nov, pp. 2579–2605, 2008.
- [52] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How powerful are graph neural networks?” in *ICLR*, 2019.
- [53] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.
- [54] Y. Zhang, Y. Zhang, D. Yan, Q. He, and Y. Yang, “Nie-gcn: Neighbor item embedding-aware graph convolutional network for recommendation,” *IEEE Transactions on Systems, Man, and Cybernetics (TSMC)*, vol. 54, no. 1, pp. 2810–2821, 2024.
- [55] H. Wang, K. Zhou, X. Zhao, J. Wang, and J.-R. Wen, “Curriculum pre-training heterogeneous subgraph transformer for top-n recommendation,” *ACM Transactions on Information Systems (TOIS)*, vol. 41, no. 1, pp. 1–28, 2023.
- [56] Y. Wang, L. Sang, Y. Zhang, and Y. Zhang, “Intent representation learning with large language model for recommendation,” *arXiv preprint arXiv:2502.03307*, 2025.