Heterogeneous Graph Contrastive Learning with Attention Mechanism for Recommendation

Ruxing Li, Dan Yang, Xi Gong

Abstract—**Existing recommendation algorithms based on heterogeneous graphs often face performance limitations due to the sparsity and nonlinearity of the heterogeneous graph structure and semantic information, which hinders the full exploitation of the association information between users and items. In order to tackle these challenges and improve the quality of user and item feature representations, a Heterogeneous Graph Contrastive Learning Recommendation algorithm based on Attention Mechanism (HAMRec) has been proposed. To enhance the robustness of graph representations, this algorithm introduces an unsupervised contrastive learning approach and utilizes attention mechanisms on top of graph neural networks to extract both local and global information from different heterogeneous graphs. Considering the varying impact of heterogeneous auxiliary information on recommendation results in real-life scenarios, HAMRec employs personalized knowledge transfer to enhance self-supervised learning. Through a large number of experiments, it has been proven that HAMRec surpasses existing baseline models in recommendation tasks, proving its effectiveness and superiority.**

Index Terms—**Graph neural network, Attention Mechanism, Graph Contrastive Learning, Personalized Information Transfer**

I. Introduction

A recommendation algorithm is an intelligent information filtering technology that provides personalized suggestions of products or services for each user based on their interests and preferences. Recommendation systems have wide applications in areas such as e-commerce, social networks, online education, etc., bringing significant value and convenience to users and businesses[1]. However, recommendation systems encounter several challenges, including data sparsity[2], cold-start problem, diversity and dynamics[3] of users and items.

In recent years, to tackle these challenges, Graph Neural Networks (GNNs)[4] have been extensively applied in recommendation algorithms to encode the interaction information between users and items existing in graph structures. The key idea of Graph Neural Networks is to aggregate neighbor information to update node features and propagate messages through graph layers. However, given the limitations of graph-based collaborative filtering methods, many heterogeneous interactions are overlooked, and only homogeneous interaction relationships are captured. In addition, traditional graph collaborative filtering methods often operate on a single user item relationship graph. In contrast, real-world recommendation scenarios contain complex graph structures with various important semantic relationships, such as connections between social relationships and item connectivity based on attributes. Therefore, many researchers had attempted to design heterogeneous graph[5] learning models based on powerful graph neural networks, which can encode their rich semantic relational information graphs into potential representation.

However, the potential representation learning ability of heterogeneous graph neural network (HGNN) models[6] is limited by sparse label information, potentially resulting in low-quality feature representations of users and items. To tackle this issue, leveraging contrastive self-supervised learning with unlabeled data is a viable approach. The fundamental concept of contrastive self-supervised learning is to generate different data views through data augmentation methods and maximize the similarity between positive sample pairs while minimizing the similarity between negative sample pairs. Contrastive learning can improve model performance stability in practical applications by enhancing feature representation quality, increasing data utilization, and reducing reliance on labeled data. Its combination with graph neural networks is known as graph contrastive learning, which has become an effective method for improving representation learning robustness. A generalized approach of graph contrastive learning is to study the consistency of feature representations from two contrasting views of graphs.

However, while some recommendation algorithms based on self-supervised heterogeneous graph contrastive learning have achieved performance improvements, they still face some challenges as follows:

- Current research tends to ignore the impact of the variety among node types on recommendations. Different node types in heterogeneous graphs may have varying importance and contribution. However, existing algorithms still have limitations in considering these inter-node type relationships. The lack of in-depth exploration of the variability of node types may lead to instability and inaccuracy in recommendation results.
- Most existing graph contrastive learning recommendation methods primarily focus on view enhancement to improve recommendation performance by increasing the

Manuscript received March 26, 2024; revised August 28, 2024. This work was supported by the General Scientific Research Project of Liaoning Provincial Department of Education (LJKMZ20220646).

Ruxing Li is a postgraduate student at School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (e-mail: lrx7156@163.com).

Dan Yang is a professor at School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (corresponding author to provide email: asyangdan@163.com).

Xi Gong is a lecturer at School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (email: askdjy05gx@163.com).

diversity of the original view. However, it has been shown that traditional view enhancement methods not only increase the model's complexity but also do not significantly improve recommendation performance.

- Different views have different levels of impact on the preference choices of different users. Some users may choose items based on their preferences, while others are often influenced by recommendations from social friends when selecting items. Therefore, it is essential to effectively utilize information from various views to explore the varying degrees of influence preferences for individuals, leading to more personalized node embeddings.

Based on the aforementioned issues, we propose a model called Heterogeneous Graph Contrastive Learning with Attention Mechanism for Recommendation (HAMRec) to address them. To encode high-order heterogeneous relationships, we leverage a HGNN to capture embeddings from three different interaction connection graphs, respectively. To better improve the model's generalization ability and stability, an effective strategy during model training is to employ data augmentation methods rather than view augmentation techniques. Then, an attention mechanism lead into HAMRec to combine different node features and obtain global node embeddings for the next layer input of the HGNN. Additionally, a cross-view information fusion network is employed to encode user and item features for final contrastive learning optimization.

This paper primarily contributes the following:

- To address data sparsity and improve the model's generalization and robustness, random noise is added for data enhancement.
- To account for the differing impacts of node types on user and item embeddings, a weighted attention mechanism is introduced to perform the fusion between nodes, resulting in node embeddings that incorporate attention mechanism for final model optimization.
- A new recommendation algorithm model called HAMRec, based on heterogeneous graph contrastive learning, is proposed to address the challenges faced by existing recommendation algorithms and improve their performance.
- Extensive experiments on public datasets Ciao, Epinions, and Yelp indicate that HAMRec enhances model performance, surpassing that of baseline models.

II. Related Work

A. Recommendation Algorithms Based on Attention Mechanism

Because of the successful application of transformers in the NLP, the attention mechanism had attracted attention in various fields. Some researchers had started to incorporate attention mechanisms into recommendation algorithms to enhance performance. Related work[7] proposed the AFM model to use an attention mechanism to solve the problem of poor feature representation. In the FM[8] model, where all feature interactions were given equal weight. DANSER[9] used two graph attention networks to capture user social similarity and item feature similarity, respectively, to further mitigate the data sparsity problem in recommendation algorithms. Researchers had also employed attention mechanism to incorporate feature interactions for downstream tasks, as seen in related work [10] and [11]. In real-world applications, object relationships often went beyond pairwise relationships, therefore, hypergraphs had received widespread attention in many practical recommendation scenarios in recent years. Related work[12] had designed a state generation module that includes gated recurrent units and attention networks to capture users' preferences over different periods, along with their historical scores.

B. Recommendation Algorithms Based on Heterogeneous Graph Neural Network

In the field of recommendation algorithms, the application of HGNNs was receiving increasing attention. Recommendation systems usually involved multiple types of entities (for example users, items, tags, etc.) and complex relationships among them. HGNN can effectively capture the heterogeneity among these entities and relationships, providing more accurate and personalized recommendations. ACKRec[13] used HGNN to integrate entities and relationships from users, items, and knowledge graphs in different domains to improve the performance of cross-domain recommendations. Many existing HGNNs used attention mechanisms to aggregate neighboring nodes, thereby enhancing the expressive and differentiation of heterogeneous nodes, as seen in related work [14] and [15]. Additionally, in related work [16], HFGN was a HGNN based on hierarchical feature graphs, which utilized multi-layer graph convolution to capture both low-order and high-order features of users and items, enhancing the accuracy and diversity of algorithms.

C. Recommendation Algorithms Based on Contrastive Learning

Contrastive learning, an unsupervised technique, enhanced the representation and discriminative power of data by capturing similarities among similar instances and differences among dissimilar ones. In the context of recommendation systems, contrastive learning effectively addressed challenges related to sparse data and biased distribution, leading to improved accuracy and diversity in recommendations. Previous studies [17] had primarily focused on enhancing session-based recommendation systems through self-supervised learning techniques and hypergraph convolutional networks. DHCN models interaction data as a hypergraph and employs self-supervised learning to enhance hypergraph modeling by maximizing mutual information between session representations obtained from two channels. Another study [18], while utilizing graph contrastive learning with knowledge transfer, simultaneously considered both interaction relationships between nodes and clustering relationships between nodes. Additionally, a proposed SSL model in related work [19] leveraged data augmentation and contrastive learning techniques to improve the quality and robustness of item representations, along with multi-task learning and knowledge distillation for enhanced model efficiency. Furthermore, attention mechanisms were employed in some research works [20] to capture multiple relationships between nodes for inter-view contrastive learning.

To address this research direction, our proposal aims to improve the effectiveness of recommendation models by incorporating an attention mechanism into a heterogeneous graph contrastive learning model.

III. Preliminaries

This section primarily outlines key symbols and definitions relevant to the paper.

In HAMRec, the user-user social graph G_{uu}, item-item similarity graph G_{vv}, and user-item interaction graph G_{uv} are built based on the raw data separately.

Definition 1. User-user social network graph G_{uu}. It is defined as $G_{uu}=\{U,\varepsilon_u\}$ where ε_u represents the set of all social relationships between users. For instance, $(u_i,u_j)\in\varepsilon_u$ indicates that users u_i and u_j can be connected through a social relationship.

Definition 2. User side information matrix A_{uu}. The paper defines the graph $G_{uu}=(V_u,E_{uu})$ to represent user side information, where E_{uu} represents the set of user side information. The adjacency matrix of the G_{uu} is denoted as $A_{uu}\in R^{m*m}$.

Definition 3. User-item interaction network graph G_{uv}. The user-item interaction network graph is defined as $G_{uv}=\{U,V,R\}$, where R is the rating matrix of users in set U on items in set V, defined as $R^{(n\times m)}=\{r_{ij}\mid i\in U, j\in V\}$. For example, the rating r_{ij} denotes the level of interaction between user u_i and item v_j.

Table I below provides specific symbol definitions:

TABLE I
STATISTICS OF THE SYMBOL

Symbol	Description
r_{ij}	the true rating of user u_i on item v_j
R	the rating matrix of users on items
G_{uu}	user-user social graph
G_{vv}	item-item similarity graph
G_{uv}	user-item interaction graph
ε_u	the set of edges in the user social network graph
ε_v	the set of edges in the item similarity network graph
d	dimensions of feature embedding for user and item
U	the set of nodes in the user social network graph
V	the set of nodes in the item similarity network graph
e_{ij}	embedding of the rating of u_i on item v_j

IV. Algorithm Framework

The proposed framework for recommendation algorithm, as shown in Figure 1, is explained in this section. HAMRec comprises four key components: 1) Initial node embeddings, where HGNN are used to learn embeddings from the user-user social graph, item-item similarity graph, and user-item interaction graph; 2) Heterogeneous information fusion and propagation, where user fusion embeddings and item fusion embeddings are used as inputs to dynamically model user preferences, item influences, and social influences using the LightGCN[21] algorithm through a node diffusion module in a recursive manner; 3) Cross-view information transfer, employing a cross-view meta-network to perform personalized information transfer from the user and item sides to obtain their final embeddings; 4) Recommendation list generation, using InfoNCE-based contrastive learning loss combined with Bayesian Personalized Ranking (BPR) loss for model optimization and final recommendation prediction.

A. Initial Node Embeddings

First, use a HGNN to learn embeddings from the user-user social graph, item-item similarity graph, and user-item interaction graph. Specifically, use the Xavier initializer to initialize node-specific embeddings and obtain the initial embedding matrices $E_u^0\in R^{i\times d}$, $E_v^0\in R^{j\times d}$ and $E_{uv}^0\in R^{(i+j)\times d}$. In this case, i represents the number of users while j represents the number of items, with d denoting the hidden dimension. Then, use a self-gating method to highlight the differences between different types of edges. The following method derives user social edge information embeddings and item similarity edge information embeddings from the initial embeddings:

$$E_{uu}^0=E_u^0*\sigma(E_u^0W+b) \tag{1}$$

$$E_{vv}^0=E_v^0*\sigma(E_v^0W+b) \tag{2}$$

Where $E_{uu}^0\in R^{i\times d}$ and $E_{vv}^0\in R^{j\times d}$ represents the initial embeddings of the user social graph G_{uu} and item similarity graph G_{vv}, respectively. The sigmoid activation function $\sigma(\bullet)$ is used, while $*$ indicates the Hadamard product, which is the element-wise multiplication of matrices. $W\in R^{d\times d}$ and $b\in R^{d\times 1}$ are the parameters that will be learned. By employing this technique of self-gating, the initial embeddings E_u^0 and E_v^0 are reweighted to obtain basic embeddings E_{uu}^0 and E_{vv}^0 that have similar semantics to E_u^0 and E_v^0 but emphasize their own features more prominently.

B. Heterogeneous Information Propagation and Fusion

1) Heterogeneous Information Propagation

In HAMRec, the initial embedding matrices E_{uu}^0、 E_{vv}^0 and E_{uv}^0 as inputs for the user-user view, item-item view, and user-item view, respectively. Then, based on graph neural network methods, propagation and prediction functions are applied on these three graphs to produce representations for both users and items. The specific method follows LightGCN, due to the increased complexity caused by nonlinear activation and feature transformation, they are discarded. Obtain the following information dissemination process:

$$e_u^{(k+1)} = \sum_{v \in N_u} \frac{1}{\sqrt{|N_u||N_v|}} e_u^{(k)} \tag{3}$$

$$e_v^{(k+1)} = \sum_{u \in N_v} \frac{1}{\sqrt{|N_u||N_v|}} e_v^{(k)} \tag{4}$$

Where $|N_u|$ and $|N_v|$ represent the neighborhood sets of the target nodes u and v respectively. To avoid embedding scale increase with graph convolution operations, symmetric normalization term $\frac{1}{\sqrt{|N_u||N_v|}}$ follow the design of standard graph convolutional neural networks. $e_u^{(k)}$ and $e_v^{(k)}$ respectively represent the embedding vectors of user u and item v during the k-th iteration propagation process. The embeddings E_{uu}^k and E_{vv}^k for the user-user social graph and item-item similarity graph are obtained similarly using the same propagation methods.

2) Data Augmentation

The crucial factor in recommendation algorithms lies in the interaction information between users and items, and perturbing them often leads to loss of critical information. However, to retain decisive information and make the representations more uniform, thus improving the model's robustness and ability to apply knowledge broadly, random noise is added to the user-user social graph and item-item similarity graph for data augmentation instead of view enhancement. The details are as follows:

$$E_{uu\_noise}^k = E_{uu}^k + \overline{N} \tag{5}$$

$$E_{vv\_noise}^k = E_{vv}^k + \overline{N} \tag{6}$$

Where $\overline{N}$ is the random noise added for data augmentation, $E_{uu\_noise}^k$ and $E_{vv\_noise}^k$ are the obtained user embeddings and item embeddings, respectively. $\left\|\overline{N}\right\|_2 = \gamma$, and γ is a hyperparameter. By adding random noise, the original data is slightly altered while preserving most of its information, which enhances the model's generalization ability. Data augmentation is applied exclusively to the training set.

3) Heterogeneous Information Propagation

By propagating heterogeneous information through multiple layers, the high-order embeddings of nodes retain the heterogeneous semantics of multi-hop connections.

The input information for each propagation is aggregated by heterogeneous relationships. The fusion of node embeddings from different graphs is achieved through a weighted attention mechanism, resulting in the generation of combined node embeddings for input into the subsequent layer. This approach takes into account the varying influence of distinct information on specific nodes. The detailed calculation procedure is outlined as follows:

$$\widehat{E}_u^{k+1} = Att(E_{uu\_noise}^{k+1}, E_u^{k+1}) \tag{7}$$

$$\widehat{E}_v^{k+1} = Att(E_{vv\_noise}^{k+1}, E_v^{k+1}) \tag{8}$$

$Att(\bullet)$ represents the fusion of nodes using a weighted attention mechanism, while $\widehat{E}_u^{k+1}$ and $\widehat{E}_v^{k+1}$ represent the fused node embeddings from different graphs. Then, the embeddings containing different heterogeneous information in $\widehat{E}_u^{k+1}$ and $\widehat{E}_v^{k+1}$ are stacked element-wise to obtain the input $\widehat{E}_{uv}^{k+1}$ for the next layer of the user-item interaction graph.

To ensure that all layers of heterogeneous information contribute to recommendations, it is necessary to further aggregate the heterogeneous information across all layers. The method for aggregating embeddings on the user-item interaction view is outlined as follows:

$$E_{u\_s} = E_u^0 + \sum_1^K \frac{E_u^k}{E_u^1 + E_u^2 + \cdots + E_u^k} \tag{9}$$

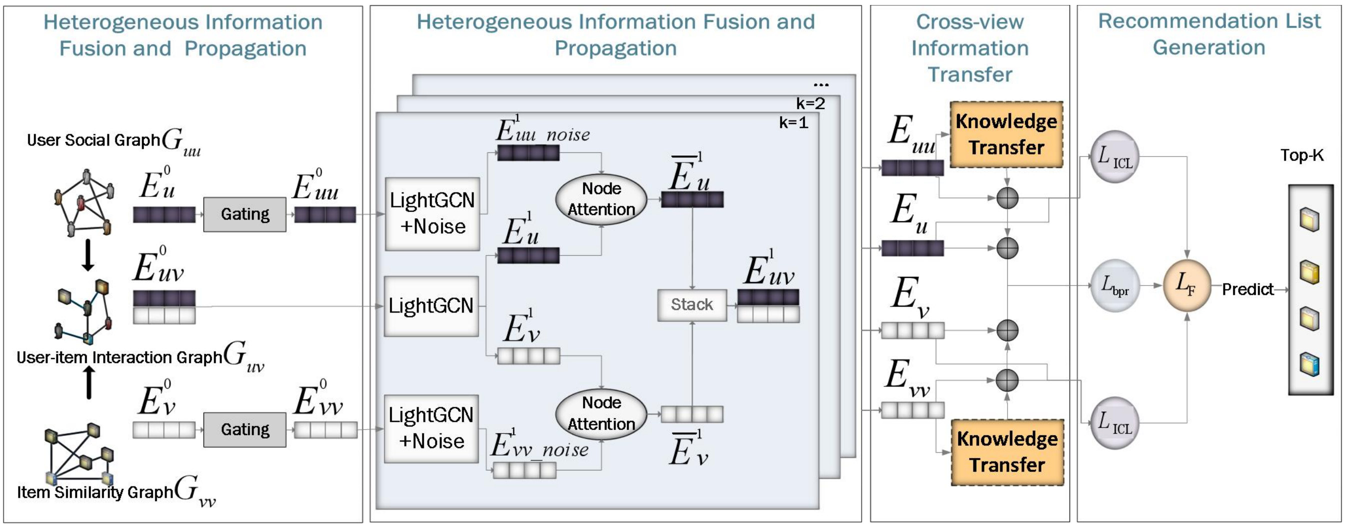

Fig. 1. Overall Framework of the HAMRec

$$E_{v\_s} = E_v^0 + \sum_{1}^{K} \frac{E_v^k}{E_v^1 + E_v^2 + \cdots + E_v^k} \tag{10}$$

The formula mentioned above represents the relationship between the outputs of all graph convolutional layers and the initial embeddings after normalization. In this case, K represents the maximum number of iterations for the graph convolutional neural network. By utilizing a consistent multi-order aggregation approach in both user-user social view and item-item similarity view, we also obtain their respective embeddings E_{uu} and E_{vv}.

C. Cross-view Information Transfer

Different views have varying impacts on the preference choices of different users. Some users choose items based on their own preferences, while others are more swayed by recommendations from their social connections. Therefore, this paper employs a cross-view information fusion network to learn different user preferences for better recommendation performance. Firstly, a parameterized feature transfer network is used to personalize the transfer of feature embeddings:

$$P_{uu}^{m1} = f_{uu}^1(M_{uu}) \tag{11}$$

$$P_{uu}^{m2} = f_{uu}^2(M_{uu}) \tag{12}$$

$f_{uu}(\bullet)$ is a fully connected feedforward neural network layer employing the PReLU activation function. P_{uu}^{m1} and P_{uu}^{m2} are personalized parameter matrices outputted by the fusion information M_{uu} through the feature transfer network. This transfer varies based on user-specific attributes, hence termed personalized information transfer. The personalized information here encompasses the embeddings E_{uu} and E_{u_s} extracted from the user-user social view and user-item interaction view. The fused information M_{uu} integrates node information from three dimensions, as follows:

$$M_{uu} = f_{fusion}(E_{uu}, E_{u\_s}, \sum_{v \in N_u} e_v) \tag{13}$$

Where $f_{fusion}(\bullet)$ represents the fusion function. By integrating information from three dimensions, the extracted fusion information can more personalized reflect the signal features of each node, enhancing the comprehensiveness and integrative nature of the cross-view fusion network. Next, a personalized connectivity function is used to construct a personalized information propagation network for connecting cross-view information:

$$\widetilde{E}_u = P(P_{uu}^{m1} P_{uu}^{m2} E_{uu}) \tag{14}$$

Where $P(\bullet)$ represents the PRelu activation function. After enhancing the embeddings using a customized personalized embedding enhancement method, and $\widetilde{E}_u$ denotes the encoded user embeddings in the interaction between users and items after personalization transfer that incorporates the user-user social view embedding information. Finally, all user embeddings are fused using a weighted method:

$$E_{u\_f} = \alpha * E_{u\_s} + (1-\alpha) * (E_{uu} + \widetilde{E}_u) \tag{15}$$

Where α is a hyperparameter used to adjust the weights of different views. The generated E_{u_f} represents the ultimate user embeddings that are employed for the purpose of recommendation. The final embeddings E_{v_f} of cross-view project information is generated similarly using the aforementioned method and it is also used for the final recommendation task.

D. Recommendation List Generation

1) Cross-View Contrastive Learning Enhancement

To mitigate the issue of overfitting to the training set and enhance its generalization ability to unseen data, HAMRec uses cross-view contrastive learning methods. This involves incorporating embeddings from auxiliary views (i.e., user-user social view embeddings E_{u_f} and item-item similarity view embeddings E_{v_f}) into contrastive learning to effectively model user-item interaction and enhance self-supervised signals. The regularized embeddings E_{uu}^a and E_{vv}^a from the auxiliary views are generated using their specific embeddings E_{uu} and E_{vv} through a personalized non-mapping function in the cross-view fusion network. The fused embeddings from auxiliary views are obtained as follows:

$$E_{u\_a} = E_{u\_f} \oplus E_{uu}^a \tag{16}$$

$$E_{v\_a} = E_{v\_f} \oplus E_{vv}^a \tag{17}$$

Where $\oplus$ denotes element-wise addition. E_{u_a} and E_{v_a} are cross-view information embeddings that integrate auxiliary view information for contrastive learning. Then, they are combined with embeddings from the user-item interaction view for enhancing contrastive learning. Building upon the recent successful application[23] of contrastive learning in recommendation algorithms, a contrastive learning loss based on InfoNCE is used to improve the representation quality and reinforce contrastive learning:

$$L_{ICL}^{(u)} = \sum_{u \in V_u} -\log \frac{\exp(s(e_{u\_a} \cdot e_u)/\tau)}{\sum_{\tilde{u} \in V_u} \exp(s(e_{u\_a} \cdot \tilde{e}_u)/\tau)} \tag{18}$$

$$L_{ICL}^{(v)} = \sum_{v \in U_v} -\log \frac{\exp(s(e_{v\_a} \cdot e_v)/\tau)}{\sum_{\tilde{v} \in U_v} \exp(s(e_{v\_a} \cdot \tilde{e}_v)/\tau)} \tag{19}$$

Where E_* represents the embedding vectors of the corresponding matrices, $S(\bullet)$ represents cosine similarity, $\tilde{e}_u$ and $\tilde{e}_v$ as negative samples with different indices are automatically identified by the temperature coefficient τ . Finally, the overall contrastive loss is obtained by combining user-side and item-side contrastive losses

$$L_{ICL}=\alpha L_{ICL}^{(u)}+\beta L_{ICL}^{(v)} \quad (20)$$

Where α and β are two hyperparameters used to modulate the weights of $L_{ICL}^{(u)}$ and $L_{ICL}^{(v)}$.

2) Recommendation Result Prediction

The prediction score, indicating the level of preference for unexplored items, is obtained by multiplying the embeddings of users and items:

$$\hat{y}=E_{u\_f}*E_{v\_f} \quad (21)$$

Where $\hat{y}$ represents the predicted preference score of the user for uninteracted items. To increase the preference score for interacted user-item pairs and decrease the score for uninteracted ones, the Bayesian Personalized Ranking (BPR) pairwise ranking loss is used to adjust the recommendation task. Its specific form is as follows:

$$L_{BPR}=\sum_{(u,i,j)\in O}-\log\sigma(\hat{y}_{u,i}-\hat{y}_{u,j}) \quad (22)$$

Where $\sigma(\bullet)$ represents the sigmoid activation function, O denotes all training interaction data, i represents positive sample indices where users have interacted, and j represents the negative sample indices where users have not interacted. Finally, combining the BPR loss and InfoNCE loss jointly optimizes the HAMRec model, and the overall loss is given by:

$$L_F=L_{BPR}+\lambda L_{ICL} \quad (23)$$

Where λ is a hyperparameter that control the weight assigned to the cross-loss.

V. Experiments

This section introduces the required datasets for experiments, evaluation metrics, and conducts extensive comparative experiments, ablation studies, and parameter experiments, followed by a detailed analysis of the experimental results.

A. Experimental Setup

1) Datasets

This study employs several datasets, they are sourced from three publicly available online review systems with social functionalities: Ciao (www.ciao.co.uk), Epinions (www.epinions.com) and Yelp (www.yelp.com). They include user ratings for various items, and their heterogeneous relationships are derived from user and item side knowledge including user social connections and item similarity relationships. Detailed information is displayed in Table II:

TABLE II
Statistical Information of the Datasets

Datasets	Ciao	Epinions	Yelp
Users	6,776	15,210	161305
Items	101,415	233,929	114852
Interaction	265,308	630,391	957923
Interaction Density	0.386%	0.177%	0.052%

2) Evaluation Metrics

The evaluation of the model's recommendation performance is conducted using two widely adopted metrics: Hit Ratio (HR@N) and Normalized Discounted Cumulative Gain (NDCG@N).

- *Hit Ratio*

Hit Ratio (HR) emphasizes the accuracy of the recommendations made by the model, primarily evaluating whether the items that users need are present in the model's recommendation list. The detailed calculation method is presented below:

$$HR=\frac{1}{S}\sum_{i=1}^{S}hit(i) \quad (24)$$

Where S denotes the quantity of items required by the user, while $hit(i)$ signifies the model's ability to accurately forecast the i-th item demanded by the user. A successful prediction is represented as 1, otherwise, it is 0.

- *Normalized Discounted Cumulative Gain*

The NDCG metric focuses on the prioritization of user-specified items in the suggested list. The computation methodology is outlined as follows:

$$NDCG=\frac{1}{S}\sum_{i=1}^{S}\frac{1}{\log_2(p_i+1)} \quad (25)$$

In situations where the requested item is absent from the recommendation list, a value of 0 will be assigned to $\frac{1}{\log_2(p_i+1)}$. Here, S represents the total number of items users have requested, while p_i indicates the final ranking of the requested item.

The metrics mentioned above are employed to assess the model's performance by considering the top 10 items in the ultimate recommendation list. These metrics have a scale from 0 to 1, where a higher score signifies better recommendation performance achieved by HAMRec.

TABLE III
Performance Comparison of HAMRec with Baselines on Three Datasets

Datasets	Metrics	DGRec	NGCF	GraphRec	HERec	MCRec	HeCo	HGT	MHCN	SMIN	HGCL	HAMRec
Ciao	HR@10	0.6653	0.6945	0.6825	0.6800	0.6772	0.6867	0.6939	0.7053	0.7108	0.7376	**0.7532**
	NDCG@10	0.4953	0.4894	0.4730	0.4712	0.4708	0.4867	0.4869	0.4928	0.5012	0.5261	**0.5389**
Epinions	HR@10	0.7650	0.7984	0.7723	0.7642	0.7630	0.7998	0.8150	0.8201	0.8179	0.8367	**0.8429**
	NDCG@10	0.5663	0.5945	0.5751	0.5495	0.5326	0.5910	0.6126	0.6158	0.6137	0.6413	**0.6497**
Yelp	HR@10	0.7950	0.8265	0.8098	0.7928	0.7869	0.8359	0.8364	0.8344	0.8478	0.8712	**0.8745**
	NDCG@10	0.5593	0.5854	0.5679	0.5612	0.5590	0.5847	0.5883	0.5799	0.5993	0.6310	**0.6356**

3) Baselines

Compare HAMRec with the following 10 baseline methods:

- DGRec[23]. Considering both graph structures simultaneously allow for a better exploration of user interests and behavioral patterns, leading to more accurate recommendations.
- NGCF[24]. Considering both the first-order and higher-order interactions between users and items, and iteratively propagating information multiple times to obtain richer representations, enables better capturing of the complex relationships between users and items.
- GraphRec[25]. Enhancing the effectiveness and personalization of recommendations by leveraging the graph structure information between users and items
- HERec[26]. utilizing the rich information within the heterogeneous graph to improve the accuracy and personalization of recommendations.
- MCRec[27]. Introducing attention mechanisms to learn node representations in heterogeneous graphs helps address the challenges of heterogeneous information fusion and learning.
- HeCo[28]. Aiming to drive technological breakthroughs in heterogeneous graph data processing through an innovative co-adversarial learning mechanism.
- HGT[15]. Introducing the self-attention mechanism from transformer to capture dependencies of nodes at both local and global levels.
- MHCN[20]. Combining meta-path and convolutional neural network methods to obtaining node representations helps address the complex relationships and information propagation issues among nodes in heterogeneous graph.
- SMIN[29]. Integrating information from multiple views into a unified representation space and capturing correlations between different views through a structural perception mechanism.
- HGCL[30]. Introducing a cross-view meta-learning neural network to implement a customized relationship collaboration knowledge learning mechanism.

4) Parameter Setting

To maintain fairness in comparisons, the Adam optimizer was utilized to optimize all approaches in this study. the learning rate established at [4e-2, 6e-2], the embedding size is adjusted between [8, 128], the neural network layers is [1, 3], the contrastive loss coefficient is set to [0.32, 0.64], The range of the batch size is defined as [1024, 8192], while the adjustment for the temperature coefficient falls within [0.2, 0.7]. Additionally, the interval specified for the dimension of low-rank matrix decomposition ranges from 1 to 5.

B. Analysis of Experimental Results

Through extensive experiments on three datasets and compares with the above methods. We obtained the experimental results shown in Table III. An examination of these results revealed that the models (MCRec, HGT) which introduce attention mechanisms to learn node representations in heterogeneous graphs, effectively address the complex relationships among data nodes in heterogeneous graph learning. This demonstrates the effectiveness of attention mechanisms in heterogeneous graph learning. On the other hand, models (HeCo, SMIN, HGCL) significantly improve performance by introducing contrastive learning between multiple views, indicating that contrastive learning is crucial for enhancing the effectiveness of recommendation algorithms models.

"The experimental findings provide strong evidence of the consistent superiority of the HAMRec model compared to all other baseline models, showcasing substantial improvements in evaluation metrics. This indicates that using attention mechanisms to fuse heterogeneous nodes and employing data augmentation with added random noise contribute significantly to the model's effectiveness. The interaction density for the Ciao dataset and Epinions are 0.386% and 0.177%, and for the Yelp dataset, it is 0.052%. This suggests that the model performs better with datasets having higher interaction densities.

TABLE IV
PERFORMANCE COMPARISON OF HAMREC WITH OTHER ABLATION METHODS

datasets	Metrics	HAMRec-a	HAMRec-n	HAMRec
Ciao	HR@10	0.7429	0.7454	**0.7532**
	NDCG@10	0.5277	0.5315	**0.5389**
Epinions	HR@10	0.8391	0.8392	**0.8429**
	NDCG@10	0.6443	0.6467	**0.6497**
Yelp	HR@10	0.8724	0.8739	**0.8745**
	NDCG@10	0.6322	0.6346	**0.6356**

C. Ablation Analysis

To assess the impact of data augmentation and the attention mechanism on the overall performance of our model, we created two variations by removing specific components from HAMRec. HAMRec-n refers to the version without random noise for data augmentation, while HAMRec-a represents the exclusion of weighted attention for node aggregation. The experimental results can be found in Table IV.

The results obtained from the experiments in Table IV provide clear evidence of a significant decrease in performance for HAMRec-n when compared to the complete model. Nevertheless, it outperforms HGCL, thus confirming the effectiveness of incorporating random noise as a data augmentation technique to enhance recommendation algorithm performance. Additionally, the ablated model HAMRec-a also performs poorly, with inferior performance compared to HAMRec-n. It shows that the contribution of node information from different views to the final node representation varies, and the attention mechanism is particularly important for enhancing model performance.

D. Analysis of Hyperparameters

Next, we will study the hyperparameters and their impact on the results of HAMRec. The detailed analysis are as follows:

1) Impact of the Loss Coefficient λ

λ is the weight parameter that connects the main loss function (BPR loss) and the contrastive learning loss function (InfoNCE loss) together. Keeping all other hyperparameters constant, λ is adjusted between 0.32 and

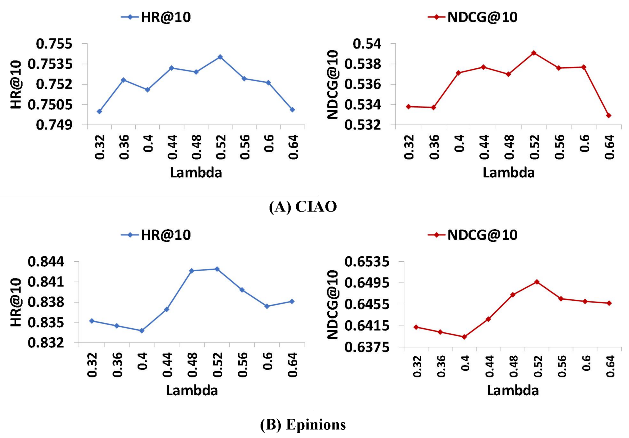

Fig. 2. Performance Comparison of different λ

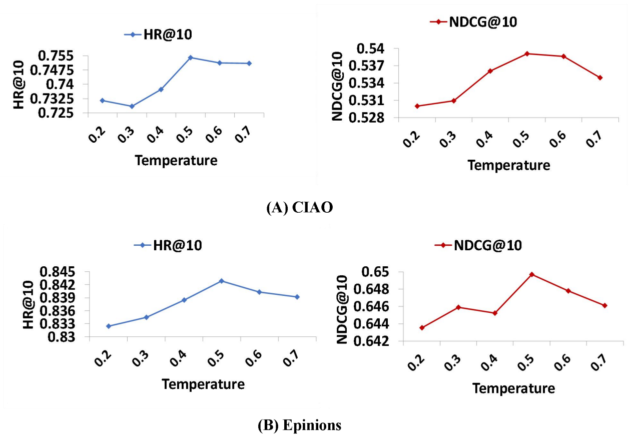

Fig. 3. Performance Comparison of different τ

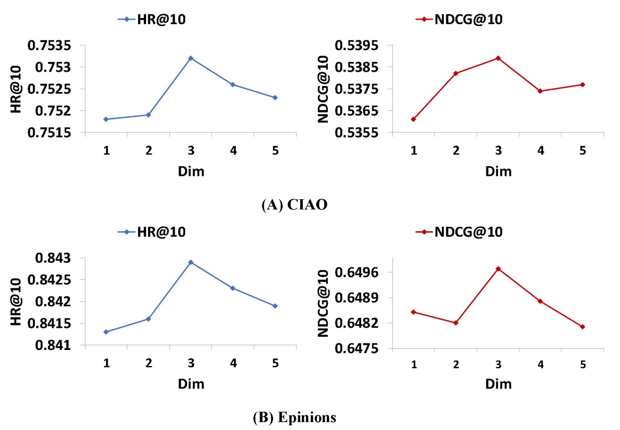

Fig. 4. Performance Comparison of different Dim

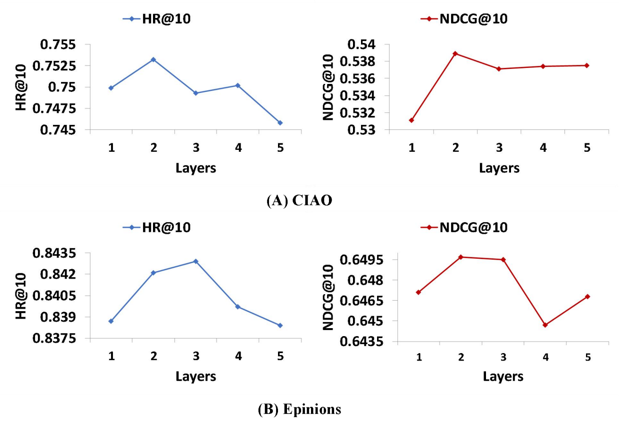

Fig. 5. Performance Comparison of different Layers

0.64. The findings from the experiment can be observed in Fig 2.

The results indicate that the recommendation model performs optimally at a λ value of 0.52, suggesting that the BPR loss function plays a crucial role as the primary loss function for the model.

2) Impact of the Temperature Coefficient τ

τ is the temperature hyperparameter of the InfoNCE loss function. Keeping all other hyperparameters constant, τ is adjusted between 0.2 and 0.7. The experimental results are shown in Fig 3.

it can be inferred that the optimal performance of the model is achieved when τ is adjusted to 0.5. However, when the temperature coefficient τ is too high or too low, the performance of the model significantly decreases. Specifically, when τ is less than 0.4, the performance of the model is quite poor.

3) Impact of the Dimension Dim

Dim is the dimension of low rank matrix decomposition. Keeping all other hyperparameters constant, Dim is adjusted between [1,5]. The experimental results are shown in Fig 4.

From the figure, we can observe that the final recommendation performance changes with the parameter Dim. When the dimension is kept at 3, HAMRec performs the best.

4) Impact of the Layers

Layers refers to the propagation layers in the graph neural network. Keeping all other hyperparameters constant, Layers is adjusted between 1 and 5. The experimental results are shown in Fig 5.

From the figure, it can be observed that in the Ciao dataset, the model performs best with 2 layers. However, in the Epinions dataset, the results are similar when Layers is 2 or 3, with the HR evaluation being slightly higher when Layers is 3.

Therefore, it is essential to select an appropriate dimension for low-rank matrix decomposition to achieve optimal recommendation performance.

VI. CONCLUSION

This paper presents a heterogeneous graph contrastive learning recommendation algorithm called HAMRec based on attention mechanisms, which effectively addresses the data imbalance issue caused by the varying influence of different users and items in heterogeneous graphs. Through attention mechanisms, the model can accurately capture and integrate node information from different graphs, providing more personalized recommendations for each user. In addition, the method of enhancing data through the addition of random noise not only improves the understanding of the model with existing data but also enhances its ability to generalize and maintain robustness when dealing with complex real-world data. Experimental results demonstrate that compared to several existing methods, HAMRec improves recommendation performance on the Ciao, Epinions and Yelp datasets, proving its effectiveness and practicality. Future work will explore further optimizations of attention mechanisms in contrastive learning and their application in a broader range of recommendation algorithms scenarios.

REFERENCES

[1] Linden, Greg, Brent Smith, and Jeremy York. "Amazon. com recommendations: Item-to-item collaborative filtering," IEEE Internet Computing, pp. 76-80, 2003.

[2] Nagarnaik P, Thomas A. "Survey on recommendation system methods," 2015 2nd International Conference on Electronics and Communication Systems (ICECS), pp. 1603-1608, 2015.

[3] Covington P, Adams J, Sargin E. "Deep neural networks for youtube recommendations," Proceedings of the 10th ACM Conference on Recommender Systems, pp. 191-198, 2016.

[4] Wang X, He X, Chua T S. "Learning and reasoning on graph for recommendation," Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 890-893, 2020.

[5] Wang D, Zhang X, Yin Y, et al. "Multi-view enhanced graph attention network for session-based music recommendation," in ACM Transactions on Information Systems, vol. 42, no.1, pp. 1-30, 2023.

[6] Jiang Y, Ma H, Zhang X, et al. "Incorporating metapath interaction on heterogeneous information network for social recommendation," Frontiers of Computer Science, vol. 18, no.1, 181302, 2024.

[7] Xiao J, Ye H, He X, et al. "Attentional factorization machines: Learning the weight of feature interactions via attention networks," arXiv preprint arXiv:1708.04617, 2017.

[8] Rendle S. "Factorization machines," IEEE International Conference on Data Mining, pp. 995-1000, 2010.

[9] Wu Q, Zhang H, Gao X, et al. "Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems," the World Wide Web Conference, pp. 2091-2102, 2019.

[10] Ouyang W, Zhang X, Li L, et al. "Deep spatio-temporal neural networks for click-through rate prediction," Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2078-2086, 2019.

[11] Cheng Y, Xue Y. "Looking at CTR Prediction Again: Is Attention All You Need?" Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1279-1287, 2021.

[12] Hou Y, Gu W, Yang K, et al. "Deep Reinforcement Learning Recommendation System based on GRU and Attention Mechanism," Engineering Letters, vol. 31, no.2, pp. 695-701, 2023.

[13] Gong J, Wang S, Wang J, et al. "Attentional graph convolutional networks for knowledge concept recommendation in moocs in a heterogeneous view," Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 79-88, 2020.

[14] Wang X, Ji H, Shi C, et al. "Heterogeneous graph attention network," the World Wide Web Conference, pp. 2022-2032, 2019.

[15] Hu Z, Dong Y, Wang K, et al. "Heterogeneous graph transformer," Proceedings of the Web Conference, pp. 2704-2710, 2020.

[16] Li X, Wang X, He X, et al. "Hierarchical fashion graph network for personalized outfit recommendation," Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 159-168, 2020.

[17] Xia X, Yin H, Yu J, et al. "Self-supervised hypergraph convolutional networks for session-based recommendation," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no.5, pp. 4503-4511, 2021.

[18] Zhang B, Yang D, Liu Y, et al. "Graph Contrastive Learning with Knowledge Transfer for Recommendation," Engineering Letters, vol. 32, no.3, pp. 477-487, 2024.

[19] Yao T, Yi X, Cheng D Z, et al. "Self-supervised learning for large-scale item recommendations," Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 4321-4330, 2021.

[20] Yu J, Yin H, Li J, et al. "Self-supervised multi-channel hypergraph convolutional network for social recommendation," Proceedings of the Web Conference 2021, pp. 413-424, 2021.

[21] He X, Deng K, Wang X, et al. "Lightgcn: Simplifying and powering graph convolution network for recommendation," Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 639-648, 2020.

[22] Xia L, Huang C, Xu Y, et al. "Hypergraph contrastive collaborative filtering," Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 70-79, 2022.

[23] Yang L, Wang S, Tao Y, et al. "Dgrec: Graph neural network for recommendation with diversified embedding generation," Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, pp. 661-669, 2023.

[24] Wang X, He X, Wang M, et al. "Neural graph collaborative filtering," Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 165-174, 2019.

[25] Fan W, Ma Y, Li Q, et al. "Graph neural networks for social recommendation," the World Wide Web Conference, pp. 417-426, 2019.

[26] Shi C, Hu B, Zhao W X, et al. "Heterogeneous information network embedding for recommendation," IEEE Transactions on Knowledge and Data Engineering, vol. 31, no.2, pp. 357-370, 2018.

[27] Hu B, Shi C, Zhao W X, et al. "Leveraging meta-path based context for top-n recommendation with a neural co-attention model," Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1531-1540, 2018.

[28] Wang X, Liu N, Han H, et al. "Self-supervised heterogeneous graph neural network with co-contrastive learning" Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 1726-1736, 2021.

[29] Lian Z, Liu B, Tao J. "Smin: Semi-supervised multi-modal interaction network for conversational emotion recognition," IEEE Transactions on Affective Computing, 2022.

[30] chen M, Huang C, Xia L, et al. "Heterogeneous graph contrastive learning for recommendation," Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, pp. 544-552, 2023.