MHGCP:Multi-View Heterogeneous Graph with Cross-View Projection for Recommendation

Xinlong Feng, Qianfang Xu(✉)*, Shaojie Tang, and Bo Xiao

Beijing University of Posts and Telecommunications, Beijing 100876, China
{fengxinlong, xuqianfang, tang_shaojie, xiaobo}@bupt.edu.cn

Abstract. Heterogeneous Graph Neural Networks excel in various recommendation scenarios by effectively modeling and leveraging diverse information. However, two key challenges remain. First, heterogeneous information, such as user-item interactions, social relationships, and category tags, is found in distinct semantic spaces. Directly merging this information can lead to semantic confusion, making it difficult for the model to differentiate between relationships and reducing recommendation accuracy. Second, each type of heterogeneous relationship contains unique semantic characteristics. Current methods often focus solely on connectivity, neglecting these unique semantics, which limits the model's ability to understand and represent heterogeneous information effectively. To address these challenges, we propose a novel approach named Multi-view Heterogeneous Graph with Cross-view Projection (MHGCP). This approach creates independent views for each heterogeneous semantic type to mitigate semantic confusion. Additionally, it introduces a cross-view projection layer that facilitates information transfer between semantic views and encodes inter-view relationships, allowing the model to indirectly capture the unique properties of each view. We tested our model on three real datasets, demonstrating superior performance. Through ablation studies and case studies, we validated the contribution of key modules in our approach to performance improvement. The implementation of the model can be found on https://github.com/fxl951677676/MHGCP.

Keywords: Recommendation system · Graph Neural Networks · Heterogeneous Graph.

1 Introduction

Recommender Systems (RS)[13, 32, 8] aim to alleviate the information overload by delivering personalized content to users. Graph Neural Networks (GNNs), through layered aggregation of neighbor information, enable nodes to capture high-order insights. Recent advancements in GNNs incorporate heterogeneous graphs, allowing systems to model diverse node and edge types. This approach enhances the representation of associations within user and item graphs, contributing to accurate and diverse recommendation services.

* Corresponding author.

The core challenge in heterogeneous graph recommendation lies in managing diverse relationships and optimizing the use of each heterogeneous information type. Existing methods, such as HERec [26] and VRKG4Rec [23], adopt random walks or encode relationship types but risk introducing confusion by oversimplifying distinct semantic spaces. As illustrated in Figure 1, We have created a two-dimensional schematic diagram to show a potential confusion in existing methods. We believe that the vector representation spaces for each heterogeneous semantic are distinct. Simple addition or average pooling during fusion can produce highly similar vectors for entities dissimilar across different semantic spaces, complicating the distinction between entities.

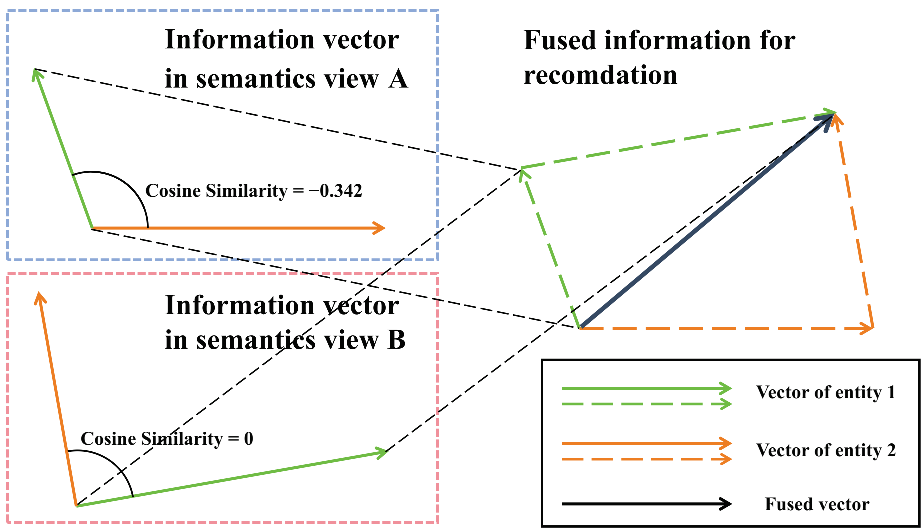


Fig. 1. Illustration of the Information Confusion Issue: The left side depicts the vector representations of two entities in semantic views A and B, while the right side illustrates the result after a simple vector addition.

To address these challenges, we introduce Multi-view Heterogeneous Graph with Cross-view Projection (MHGCP), inspired by two-stream networks [28]. MHGCP adopts a divide-and-conquer approach, creating separate views for each heterogeneous semantic to reduce confusion in information propagation. Within each view, the aggregation process consolidates neighbor information for nodes, maintaining distinct semantic characteristics. To bridge these views, we designed a cross-view projection layer, introducing trainable parameters and a learnable transfer matrix that encodes relationships between semantic views, capturing each view's intrinsic meaning and enhancing recommendation accuracy.

In summary, this paper makes the following contributions:

– We proposed a method to handle heterogeneous information by creating separate views for each semantic type and aggregating information within them, reducing confusion during relationship aggregation.
– We introduced a cross-view projection module with trainable parameters to encode relationships between views, capturing heterogeneous semantic characteristics and enabling cross-view information transformation.
– Extensive experiments on real datasets demonstrate MHGCP's superior performance over baselines, with each component's contribution validated for performance improvement.

2 Related Work

GNN-based Recommendation GNN-based methods outperform traditional approaches by aggregating high-order information through stacked layers. Their proficiency in handling structured data has made GNNs pivotal in recommendation system research. For example, GC-MC [2] treats recommendation as link prediction on graphs, HGNN [11] captures high-order relationships in hypergraphs, And GAN[30] introduces attention for node classification in graph data. While classical GNNs excel in recommendations, they struggle with diverse information. Heterogeneous graph models enhance recommendation accuracy by handling various node types and relationships, motivating our adoption of GNN-based approaches on heterogeneous graphs for recommendation.

Heterogeneous Graph Recommendation In recommendation scenarios, diverse relationships among entities like users and items are naturally represented as networks. Traditional GNNs focus on homogeneous networks, neglecting heterogeneity in entities and relationships. To capture comprehensive structural and semantic information, researchers introduced heterogeneous information networks (HINs)[27]. Early methods like DeepWalk [24] used unsupervised learning on graph paths, leading to further methods like metapath2vec [9] and HIN2Vec [12] for heterogeneous graphs. Heterogeneous graph neural networks, such as HAN [35], utilize meta-path modeling to capture high-level semantics.Recent works [15, 3, 19] apply dual-layer aggregation along metapaths. Our approach similarly models meta-paths to learn user-item embeddings, capturing heterogeneous semantic information for recommendation.

Social Recommendation User behavior is influenced by social connections [6], and social recommendation models leverage this influence. For instance, SMIN[22] introduces a heterogeneous graph neural network using meta-relation. HGCL[5] proposed a novel contrastive learning framework for heterogeneous graph. TGRec[1] introduces a time-augmented graph model for social recommendations. Social recommendation operates within the framework of heterogeneous graph recommendation. These approaches enhance user-item interaction representation, effectively utilizing social information. While our approach shares this emphasis, what distinguishes it is the equal importance given to various heterogeneous information types. Our proposed model framework aims to leverage diverse heterogeneous information sources to enhance recommendation performance.

3 METHODOLOGY

3.1 Preliminaries

First, we need introduce key symbols and definitions related to the method. Consider a recommendation scenario involving users $\mathcal{U} = \{u_1, ...u_i, ...u_I\}$ and items $\mathcal{V} = \{v_1, ...v_j, ...v_J\}$, where the interactions between users and items are denoted as $G_{ui} = \{(u_i, v_j)|u \in \mathcal{U}, v \in \mathcal{V}\}$, if the interaction between u_i and v_j is observed, then $(u_i, v_j) = 1$; and $(u_i, v_j) = 0$ otherwise. Meanwhile, we have also defined subgraphs $G_{uu} = \{(u_i, u_j)|u_i, u_j \in \mathcal{U}\}$ and $G_{ii} = \{(v_i, v_j)|v_i, v_j \in \mathcal{V}\}$. Similar to the previous definition, taking G_{uu} as an example, if two users u_i and u_j have some form of heterogeneous relationship, then $(u_i, u_j) = 1$, and vice versa. Our recommendations are based on the information contained in $G = \{G_{ui}, G_{ii}, G_{uu}\}$. It's important to add that our model still has scalability and unexplored generality. If there are additional heterogeneous relationships between users or items, we can establish more subgraphs to leverage this supplementary information.

Task Description. With the above definitions, We can define this recommendation scenario as: **Input** the user-item information Graph G. **Output** a trained model that can predict unobserved U-I interactions (u_i, v_j).

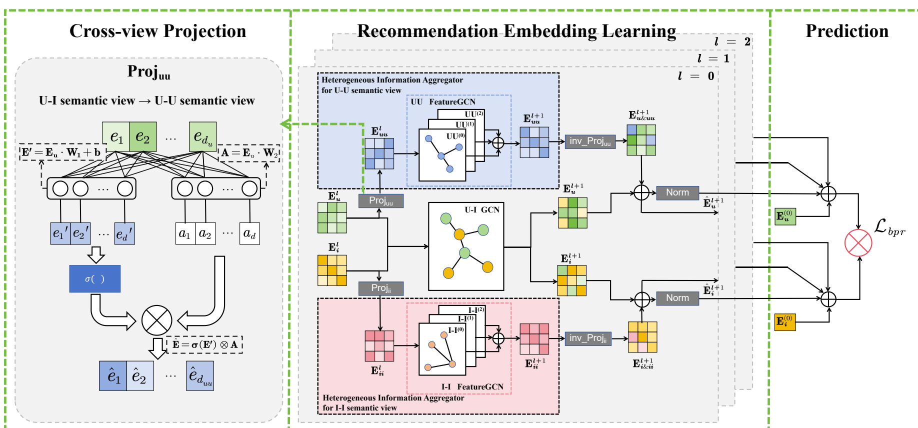


Fig. 2. Overview of Model

3.2 The process of message propagation and aggregation.

Aggregation in interaction semantics views We begin by initializing embeddings e_u and $e_i \in \mathbb{R}^d$, corresponding to user and item IDs, utilizing the Xavier initializer[14]. Here, d denotes the hidden dimension. For the sake of model simplicity, we employed a straightforward graph convolutional network structure for the fusion of homogeneous semantics. Specifically, we refine the user embeddings with the message propagation as follows:

$$\begin{aligned}\mathbf{e}_u^{l+1} &= \mathrm{AGG}\left(\mathbf{e}_u^l, \left\{\mathbf{e}_i^l : i \in \mathcal{N}_u\right\}\right) \\ &= \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} \mathbf{e}_i^l\end{aligned} \tag{1}$$

where $e_i^l, e_u^l \in \mathbb{R}^d$ denotes the embedding vectors of user u and item i in the l-th layer. $\mathcal{N}_u$ denote the neighbors of target user u in the graph G_{ui}, we used self-connections, meaning we defined u as its own neighbor. The definition of $\mathcal{N}_i$ is analogous. AGG$(\cdot)$ denotes our message aggregation function. We aggregate neighbors' information of each node. $|\mathcal{N}_i|$represents the number of neighbors of a node. We introduce the neighbor count in the denominator to prevent some less popular items from being overly influenced by popular neighbors during the aggregation. Following the approach of [17], we also remove feature transformations and non-linear activation calculations during the message propagation process.

Aggregation in heterogeneous semantics views For messages in heterogeneous semantic views, we employed the similar message propagation mechanism as the aggregation of interaction semantics view. We use social heterogeneous semantic information as an example to demonstrate the process of heterogeneous information propagation, as follows:

$$\begin{aligned}\mathbf{e}_{uu}^{k+1} &= \mathrm{AGG}\left(\mathbf{e}_{uu}^k, \left\{\mathbf{e}_{u'}^k : u' \in \mathcal{N}_{uu}\right\}\right) \\ &= \sum_{u' \in \mathcal{N}_{uu}} \frac{1}{\sqrt{|\mathcal{N}_{u'}|}\sqrt{|\mathcal{N}_{uu}|}} \mathbf{e}_{u'}^k\end{aligned} \tag{2}$$

where $\mathbf{e}_{uu}^k, \mathbf{e}_{u'}^k$ denote the k-th layer's embedding vectors in U-U semantic view. And $\mathcal{N}_{uu}$ denote the neighbors of target user in the graph G_{uu}. Unlike the U-I interaction view, the connections here do not represent interaction, but instead describe a type of heterogeneous relationship; thus, aggregation in the heterogeneous view is actually done by identifying nodes that have a heterogeneous relationship with the central node and aggregating the information of these neighboring nodes. In graph neural networks, the correlation between high-order neighborhood information and the central node is not as strong as that of the central node's first-order neighbors, a problem exacerbated in heterogeneous views. Therefore, we designed a weighting mechanism to integrate results from each layer, generating the final heterogeneous view embedding representation, as follows:

$$\mathbf{e}_{uu} = \sum_{k=0}^{K} \alpha_k \mathbf{e}_{uu}^{(k)} \tag{3}$$

where the hyperparameter k is used to control the number of layers in the heterogeneous GCN. And α_k represents a set of weight values used to weigh the results from each layer, which are computed through softmax from a set of trainable parameters. By introducing this set of weights, we enable the model to adaptively choose the weights for information from each layer. This set of value can also be treated directly as hyperparameters or same value, model can still achieve decent performance in most scenarios. Treating it as hyperparameters reduces the complexity of the model, but as a trade-off, it may sacrifice some performance and generalization.

Cross-view Projection Traditional approaches in heterogeneous graphs often use direct average pooling of information generated from various semantic views[21]. This can lead to the confusion of embeddings from different semantics. We design a semantic gating mechanism to achieve cross-view information projection employing a self-weighted projection to transform embeddings into different semantic facet while introducing non-linearity to enhance expressive power.

For the sake of keeping the formula concise, we introduce $\mathbf{E}$ to denote the set of all embeddings requiring semantic projection. As an example, we will introduce the process of semantic space transformation projection from U-I interaction semantic view to U-U semantic view. The projection processes of other semantic spaces are similar. To complete the semantic transformation projection process, we first need to calculate the preliminary transformed but unweighted embeddings denoted as $\mathbf{E}'$, and the process is as follows:

$$\mathbf{E}' = \mathbf{E}_u \cdot \mathbf{W}_1 + \mathbf{b} \tag{4}$$

where $\mathbf{E}_u$ is the embeddings of users in U-I semantic space. $\mathbf{W}_1 \in \mathbb{R}^{d_{\mathrm{u}} \times d_{\mathrm{uu}}}$ and $\mathbf{b} \in \mathbb{R}^{d_{\mathrm{uu}} \times 1}$ are trainable transformation matrix and bias. d_{u} and d_{uu} are hyperparameters set before training process. It's worth noting that due to the presence of this transformation matrix, we can use different vector dimensions in various semantic views. Then we need to obtain the self-weighting coefficients $\mathbf{A}$ through the following process:

$$\mathbf{A} = \mathbf{E}_u \cdot \mathbf{W}_2 \tag{5}$$

where $\mathbf{W}_2 \in \mathbb{R}^{d_{\mathrm{u}} \times d_{\mathrm{uu}}}$ is another trainable transformation matrix. To reduce the number of trainable parameters, we did not introduce bias in this process. After we get intermediate variables $\mathbf{E}'$ and $\mathbf{A}$.we implement the projection is as follows:

$$\mathbf{E}_{uu} = \sigma\left(\mathbf{E}'\right) \otimes \mathbf{A} \tag{6}$$

where $\otimes$ denotes element-wise multiplication operation. And $\sigma(\cdot)$ denotes the sigmoid activation function. We tried three activation functions, TanH, LeakyReLU[16], and ELU[7]. The choice of activation function can affect the speed of convergence, but it doesn't have a significant impact on the model's performance.

3.3 Layer Combination

After going through the series of message-passing operations mentioned above, we obtain the aggregated information. Next, we combine the information obtained from each layer of the network for the final prediction. As an example of the fusion of user embeddings, we first combine the information from the U-I interaction semantic view and the U-U heterogeneous semantic view, as shown follows:

$$\hat{\mathbf{E}}_u^l = \alpha_u \cdot \mathbf{E}_u^l + (1 - \alpha_u) \cdot \mathbf{E}_{u\&uu}^l \tag{7}$$

where $\hat{\mathbf{E}}_u^l$ denotes aggregated user embeddings in U-I interaction semantic view of layer l. $\hat{\mathbf{E}}_{u\&uu}^l$ denotes projected embeddings from U-U heterogeneous semantic view to U-I interaction semantic view. α_u is a weighting coefficient. It can be defined as a hyperparameter, or it can be designed as a trainable parameter for the model to learn. For the embeddings of each layer, we use the following normalization process to prevent the magnitude of embedding vectors from becoming too large:

$$NORM(\mathbf{E}) = \frac{\mathbf{E}}{\|\mathbf{E}\|_2} \tag{8}$$

where $\|\mathbf{E}\|_2$ is L_2-norm of embedding matrix $\mathbf{E}$. Then, we combine the embeddings obtained from each layer, as follows:

$$\hat{\mathbf{E}}_u = \mathbf{E}_u^0 + \sum_{l=1}^{L} NORM\left(\hat{\mathbf{E}}_u^l\right) \tag{9}$$

where L is a hyperparameter that defines the maximum number of layers in the model. And $\hat{\mathbf{E}}_u$ represents the final user embedding output, which is used for the final recommendation prediction.

3.4 Model Predictions

With fused embeddings $\hat{\mathbf{E}}_u$ and $\hat{\mathbf{E}}_i$, We can then calculate the predicted score for a specific user u and a specific item i using the following formula:

$$\hat{y}_{ui} = \mathbf{e}_u^T \mathbf{e}_i, \tag{10}$$

where $\hat{y}_{ui}$ denotes the final score of unobserved U-I interactions (u_i, v_j). We use these scores for model training and subsequent experimental evaluation.

3.5 Optimization Objectives

Our model is trained through backpropagation using two components of the loss function. We follow the work [25] and adopt Bayesian Personalized Ranking

(BPR) as our loss function. Additionally, to alleviate overfitting, we introduced a regularization loss. For all users, we minimize prediction loss as follows:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{bpr} + \mathcal{L}_{L_2} \\ &= -\sum_{u=1}^{I} \sum_{i \in \mathcal{N}_u} \sum_{j \notin \mathcal{N}_u} \ln \sigma \left(\hat{y}_{ui} - \hat{y}_{uj} \right) + \lambda_1 \left\| \mathbf{E} \right\|_2 + \lambda_2 \left\| \mathbf{W} \right\|_2\end{aligned} \tag{11}$$

where $\sigma(\cdot)$ denotes sigmoid function, $\|\mathbf{E}\|_2$ denotes the norm of the user along with the selected positive and negative samples, and $\|\mathbf{W}\|_2$ denotes the norm of all transformation matrix in our projection process. λ_1 and λ_2 denote hyperparameters to determine the weight of the regularization term, respectively.

4 EVALUATION

In this section, to validate the effectiveness and generality of MHGCP, we conducted a series of experimental tests. The experimental results answer the following research questions: **(RQ1)**: How does the performance of MHGCP compare to other state-of-the-art methods?**(RQ2)**: How do the key components of building MHGCP contribute to the model's performance?**(RQ3)**: How do hyperparameters affect the model's performance?**(RQ4)**: How MHGCP addresses the issue of semantic confusion and provides a visual explanation?

4.1 Experimental Settings

Datasets Descriptions. We conducted experiments on three public real-world datasets. We presented the statistical data of the datasets in Table 1 and provided a detailed description of the datasets as follows:

Table 1. Statistical information of the datasets.

dataset	User #	Item #	Interaction #	Density
CiaoDVD	6776	101415	271573	0.0395%
Epinions	15210	233929	644715	0.0181%
Yelp	161305	114852	1118645	0.0060%

- **Ciao and epninions**[1]: These are two benchmark recommendation datasets collected from online review systems, containing user rating behaviors for different items. Heterogeneous relationships were generated from the included user and item side information, such as user trust relationships and item categorization information.

[1] https://www.cse.msu.edu/tangjili/datasetcode/truststudy.html

- **Yelp**[2]: This data is collected from Yelp platform. The dataset consists of user-generated reviews and ratings for businesses and services. And it includes heterogeneous relationships, such as user social relationships, place rating behaviors, and business attributes.

Compared Baselines. The evaluation metrics we used are HR@10 and NDCG@10. For the baseline models, we collected part of the data from other research papers and conducted additional experiments to ensure that other models were performing at a reasonably high level. We compared our approach with the following state-of-the-art baselines:

- **DiffNet**[37]: It transforms social recommendations into a heterogeneous graph by incorporating social and interest networks, modeling high-order user influence and interests.
- **SAMN**[4]: They develop a dual-stage attention model using a memory network to capture varying levels of social influence on user-item interactions.
- **DGRec**[29]: This social recommendation system combines recurrent neural networks with graph attention layers to capture users' dynamic interests and social influence.
- **NGCF**$_+$[34]: This is a collaborative filtering model using graph convolution to integrate heterogeneous information for message passing.
- **KGAT**[33]: It integrated item knowledge relationships via a graph attention mechanism to model entities and relationships.
- **MKR**[31]: It is a multi-task and knowledge-aware recommendation system leveraging a knowledge graph for user-item interaction encoding.
- **GraphRec**[10]: This framework models the social and interaction graph to capture the heterogeneity of relationships by using graph attention to facilitate message passing and information fusion.
- **HERec**[26]: It utilizes meta-path-based random walks to encode heterogeneous information, using the obtained embeddings containing heterogeneous semantics as input to neural networks.
- **HGT**[18]: It introduces heterogeneous mutual attention and temporal encoding to maintain vertex and edge type representations and capture network dynamics.
- **SMIN**[22]: They propose a heterogeneous graph neural network that models user-item relationships across multiple semantic dimensions, using mutual information maximization between local and global features.
- **HeCo**[36]: It employs a cross-view contrastive mechanism to learn node embeddings from two views (network pattern and meta-path view), capturing local and high-order structures. Through cross-view contrastive learning and a view mask mechanism, the two views mutually supervise each other.
- **RecDiff**[20]: It is a diffusion-based social denoising framework that mitigates the impact of noisy social ties by performing multi-step noise diffusion and removal in the hidden space, enhancing recommendation accuracy and training efficiency.

[2] https://www.yelp.com/dataset/download

Hyperparameter Settings. We implement MHGCP in PyTorch, optimizing it with the Adam optimizer. Training is conducted with a batch size of 8192. The learning rate is tuned within [0.05, 0.052, 0.055, 0.06] and decays by 0.97 per epoch. For regularization, the BPR term coefficient is varied between [0.02, 0.025, 0.03, 0.04, 0.05, 0.06], and the projection layer coefficient is set to 0.01. The hidden dimensionality for the U-I interaction and heterogeneous semantic views is selected from [16, 32, 64, 128]. In heterogeneous view aggregation, we choose 1 to 2 graph neural layers, with a depth of 2 for the representation learning layers. Additional hyperparameter details are provided in our source code, and their sensitivity is analyzed in Section 4.4.

4.2 Performance Comparison (RQ1)

Table 2. Recommendation performance of different methods in terms of HR@10 and NDCG@10.

Dataset	Metric	DiffNet	SAMN	DGRec	$NGCF_+$	KGAT	MKR	GraphRec	HERec	HGT	SMIN	HeCo	RecDiff	MHGCP
CiaoDVD	HR	0.6747	0.6576	0.6653	0.6945	0.6601	0.6793	0.6825	0.6800	0.6939	0.7108	0.6867	0.7214	**0.7322**
	NDCG	0.4636	0.4561	0.4593	0.4894	0.4512	0.4589	0.4730	0.4712	0.4869	0.5012	0.4867	0.5145	**0.5312**
Epinions	HR	0.7699	0.7592	0.7603	0.7984	0.7510	0.7647	0.7723	0.7642	0.8150	0.8179	0.7998	0.8317	**0.8357**
	NDCG	0.5702	0.5614	0.5668	0.5945	0.5578	0.5669	0.5751	0.5495	0.6126	0.6137	0.5910	0.6439	**0.6522**
Yelp	HR	0.8048	0.7910	0.7950	0.8265	0.7881	0.8005	0.8098	0.7928	0.8364	0.8478	0.8359	0.8476	**0.8547**
	NDCG	0.5670	0.5516	0.5593	0.5854	0.5501	0.5635	0.5679	0.5612	0.5883	0.5993	0.5847	0.6040	**0.6293**

Table 2 presents the experimental statistics of our experiments with MHGCP and other state-of-the-art baselines on different datasets. We outline the key observations as follows: (1). Across all scenarios, MHGCP consistently outperforms other models in both NDCG@10 and HR@10 metrics, underscoring the efficacy of our approach. Notably, our model exhibits a substantial improvement in NDCG, highlighting its superior ranking capability in item recommendations. We attribute this outstanding performance to our approach of separately modeling heterogeneous semantic information through multi-view techniques and fusing different view information through cross-view projection. (2). Building on heterogeneous information and methods aimed at modeling high-order heterogeneous semantics, such as SMIN and HeCo (including our model), generally result in improved performance. This aligns with our research direction, emphasizing the importance of appropriately modeling heterogeneous information to boost model performance. (3). Despite utilizing fewer heterogeneous semantic information sources compared to tested versions of SMIN and HeCo, our current MHGCP model demonstrates further performance improvement. This suggests untapped potential in developing heterogeneous auxiliary information. While many knowledge graphs and related works aim to introduce more nodes and auxiliary information, our research underscores that better modeling of heterogeneous information is an equally promising avenue to enhance recommendation performance.

4.3 Ablation Study of MHGCP Framework (RQ2)

In order to investigate the impact of different components of the MHGCP model on its performance, we conducted the following ablation experiments and designed various MHGCP model variants:

- ***MHGCP-w/o-uu***: In this variant, we removed heterogeneous information from the u-u semantic view, refraining from enhancing user embeddings through heterogeneous semantics. This was done to examine the impact of user heterogenous semantics on recommendations.
- ***MHGCP-w/o-ii***: Similar to MHGCP-w/o-uu, in this variant, we exclude heterogeneous semantic information from the i-i semantic view, refraining from assisting in the training and prediction of project embeddings. This is done to examine the role of heterogeneous semantics for projects.
- ***MHGCP-w/o-proj***: In this variant, we remove the cross-view projection module and use a simple weighted average as a replacement. Our aim is to demonstrate the importance of the cross-view projection module for the model.
- ***MHGCP-w/o-heteagg***: In this variant, we removed the aggregation process within the heterogeneous semantic view and replaced it with a simple single-layer GCN for aggregating heterogeneous semantic information. Our intention was to explore the impact of our heterogeneous semantic aggregation process on the model.

Table 3. Ablation Study of MHGCP Framework.

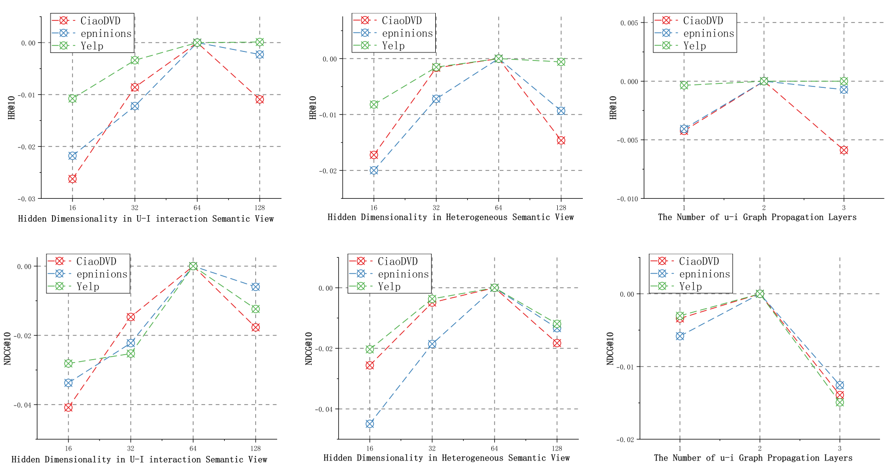
model	CIaoDVD		epninions		Yelp	
Metric	HR	NDCG	HR	NDCG	HR	NDCG
w/o-uu	0.7124	0.5062	0.8181	0.6137	0.8489	0.6065
w/o-ii	0.7103	0.4974	0.8197	0.6129	0.8493	0.6066
w/o-proj	0.7188	0.5083	0.8119	0.6032	0.8420	0.5884
w/o-heteagg	0.7200	0.5125	0.8114	0.6051	0.8446	0.5962
MHGCP	**0.7322**	**0.5312**	**0.8357**	**0.6522**	**0.8547**	**0.6293**

We evaluated the performance of the aforementioned variants as well as MHGCP on three experimental datasets. The results are shown in Table 3. Compared to these four variants, MHGCP consistently achieves the best performance. Through an analysis of the experiments, we summarized the following key findings: (1). Both variants, w/o-uu and w/o-ii, exhibit inferior performance compared to MHGCP across all datasets. This aligns with the intuitive notion that leveraging heterogeneous semantic information contributes to improved model performance. (2). A performance gap exists between the w/o-proj variant and MHGCP. We attribute this difference to the projection module, facilitating the transformation of node information between distinct semantic views. This mechanism reduces semantic confusion during information fusion and indirectly encodes intrinsic properties of heterogeneous semantics. Such encoding reveals more exploitable information, thereby contributing to the superior performance of MHGCP. (3). MHGCP outperforms w/o-heteagg, suggesting the presence of exploitable high-order neighbor information in heterogeneous relations. The exploration of this information in heterogeneous relations significantly contributes to enhancing the recommendation system's performance.

4.4 Hyperparameter Analysis (RQ3)

In this section, we analyzed the impact of vector dimensions and the number of GNN layers on the performance of the MHGCP model. Specifically, we conducted the following hyperparameter experiments, and the results are shown in Figure 3. To consolidate the experimental data into the same chart, we set the y-axis as the percentage improvement or decline relative to the performance of MHGCP.

Fig. 3. Hyperparameter Analysis of MHGCP.



- ***Hidden State Dimensionality in U-I Semantic View***: We adjust the dimension of the u-i semantic view vector between 16 and 128. observing that model performance initially increased but then saturated or declined. Increasing the embedding dimension can enhance the model's expressive power, but excessively large dimensions may lead to overfitting even with our regularization.
- ***Hidden State Dimensionality in Heterogeneous Semantic View***: Although the model has the capability to adjust the dimensions individually for each heterogeneous semantic, considering the model's complexity and the challenges of testing, we uniformly adjusted the dimensions of heterogeneous semantic view vectors within the range of 16-128. Performance trends mirrored those of the u-i semantic view.

– ***The Number of u-i Graph Propagation Layers***: We tested the number of graph layers ranging from 1 to 3. The results show an overall trend of increasing and then decreasing performance, with optimal performance observed at two layers. This indicates that aggregating higher-order neighbor aggregation improves recommendations, but excessive layers cause oversmoothing, which means vectors tend to converge to similarity after multi-layer aggregation. Although we introduced self-connections to mitigate this problem, but it can not fully prevent performance decline from too many layers.

4.5 Case Study (RQ4)

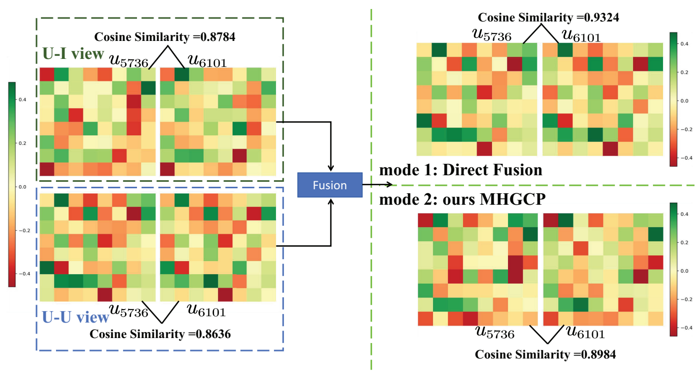


Fig. 4. Case study on CiaoDVD dataset.

In this subsection, we take users u_{5736} and u_{6101} to demonstrate how MHGCP reduces semantic confusion through visualizing their embeddings. For comparison, we introduce MHGCP-lf (late fusion), a variant that disables information interaction between views and instead performs average pooling and regularization at the end. We extract embeddings from the U-I interaction and U-U heterogeneous views for both MHGCP and MHGCP-lf. Since their embeddings are only related to their respective initial embeddings and the subgraphs of their input view, their embeddings can represent the model's learning results for each view. We visualize the embeddings of u_{5736} and u_{6101} for the U-I interaction view, U-U heterogeneous view, directly averaged pooling embeddings, and MHGCP embeddings. The visualization represents each dimension's value with colors, as shown in Figure 4.

To quantify similarity, we calculate cosine similarity between the embeddings of u_{5736} and u_{6101} across different views. In the U-I and U-U views, similarities are 0.8784 and 0.8636, respectively, while direct average pooling results in a higher similarity of 0.9324. This indicates that during aggregation, there was a semantic confusion issue, causing the fused vectors to become more similar compared to before fusion, making it difficult for the model to distinguish. Compared to the MHGCP-lf variant, it reduced this similarity to 0.8984, indicating that MHGCP alleviates semantic confusion issues during aggregation. We attribute this alleviation of confusion to the role of the projection layer, which, by projecting vectors, transfers embeddings to the vector space corresponding to the semantic view, thus avoiding confusion caused by directly adding embeddings from different spaces.

5 CONCLUSION

This study delves into challenges associated with heterogeneous graph neural networks, specifically addressing issues related to heterogeneous information confusion and the oversight of intrinsic information within heterogeneous relationships. To tackle these challenges, we introduce an innovative heterogeneous graph learning architecture named MHGCP.

MHGCP adopts a divide-and-conquer approach, employing a multi-view strategy to independently manage diverse heterogeneous semantics. Simultaneously, it incorporates a cross-view projection layer to encode semantic relationships between views, effectively capturing the inherent meanings of distinct heterogeneous semantics indirectly. Through comprehensive experiments, we validated the efficacy of MHGCP and conducted a thorough analysis of the roles played by its key modules in each experimental scenario.

Our findings suggest that MHGCP serves as a pioneering framework for resolving challenges associated with heterogeneous information. Several key modules within the framework exhibit potential for further optimization. It is noteworthy that our model demonstrates strong scalability. Simultaneously, research into aspects such as automatic selection of dimensions and layers can equally serve as prospective directions for future studies.

References

1. Bai, T., Zhang, Y., Wu, B., Nie, J.Y.: Temporal graph neural networks for social recommendation. In: 2020 IEEE International Conference on Big Data (Big Data). pp. 898–903. IEEE (2020)
2. Berg, R.v.d., Kipf, T.N., Welling, M.: Graph convolutional matrix completion. arXiv preprint arXiv:1706.02263 (2017)
3. Chang, J., Gao, C., He, X., Jin, D., Li, Y.: Bundle recommendation with graph convolutional networks. In: Proceedings of the 43rd international ACM SIGIR conference on Research and development in Information Retrieval. pp. 1673–1676 (2020)
4. Chen, C., Zhang, M., Liu, Y., Ma, S.: Social attentional memory network: Modeling aspect-and friend-level differences in recommendation. In: Proceedings of the twelfth ACM international conference on web search and data mining. pp. 177–185 (2019)

5. Chen, M., Huang, C., Xia, L., Wei, W., Xu, Y., Luo, R.: Heterogeneous graph contrastive learning for recommendation. In: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining. pp. 544–552 (2023)
6. Cialdini, R.B., Goldstein, N.J.: Social influence: Compliance and conformity. Annu. Rev. Psychol. **55**, 591–621 (2004)
7. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289 (2015)
8. Deng, Y.: Recommender systems based on graph embedding techniques: A review. IEEE Access **10**, 51587–51633 (2022)
9. Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: Scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 135–144 (2017)
10. Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., Yin, D.: Graph neural networks for social recommendation. In: The world wide web conference. pp. 417–426 (2019)
11. Feng, Y., You, H., Zhang, Z., Ji, R., Gao, Y.: Hypergraph neural networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 3558–3565 (2019)
12. Fu, T.y., Lee, W.C., Lei, Z.: Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 1797–1806 (2017)
13. Gao, C., Zheng, Y., Li, N., Li, Y., Qin, Y., Piao, J., Quan, Y., Chang, J., Jin, D., He, X., et al.: A survey of graph neural networks for recommender systems: Challenges, methods, and directions. ACM Transactions on Recommender Systems **1**(1), 1–51 (2023)
14. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp. 249–256. JMLR Workshop and Conference Proceedings (2010)
15. Gong, J., Wang, S., Wang, J., Feng, W., Peng, H., Tang, J., Yu, P.S.: Attentional graph convolutional networks for knowledge concept recommendation in moocs in a heterogeneous view. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. pp. 79–88 (2020)
16. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
17. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. pp. 639–648 (2020)
18. Hu, Z., Dong, Y., Wang, K., Sun, Y.: Heterogeneous graph transformer. In: Proceedings of the web conference 2020. pp. 2704–2710 (2020)
19. Ji, H., Zhu, J., Wang, X., Shi, C., Wang, B., Tan, X., Li, Y., He, S.: Who you would like to share with? a study of share recommendation in social e-commerce. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 232–239 (2021)
20. Li, Z., Xia, L., Huang, C.: Recdiff: Diffusion model for social recommendation (2024)
21. Liu, J., Shi, C., Yang, C., Lu, Z., Philip, S.Y.: A survey on heterogeneous information network based recommender systems: Concepts, methods, applications and resources. AI Open **3**, 40–57 (2022)

22. Long, X., Huang, C., Xu, Y., Xu, H., Dai, P., Xia, L., Bo, L.: Social recommendation with self-supervised metagraph informax network. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 1160–1169 (2021)
23. Lu, L., Wang, B., Zhang, Z., Liu, S., Xu, H.: Vrkg4rec: Virtual relational knowledge graph for recommendation. In: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining. pp. 526–534 (2023)
24. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710 (2014)
25. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)
26. Shi, C., Hu, B., Zhao, W.X., Philip, S.Y.: Heterogeneous information network embedding for recommendation. IEEE Transactions on Knowledge and Data Engineering **31**(2), 357–370 (2018)
27. Shi, C., Li, Y., Zhang, J., Sun, Y., Philip, S.Y.: A survey of heterogeneous information network analysis. IEEE Transactions on Knowledge and Data Engineering **29**(1), 17–37 (2016)
28. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. Advances in neural information processing systems **27** (2014)
29. Song, W., Xiao, Z., Wang, Y., Charlin, L., Zhang, M., Tang, J.: Session-based social recommendation via dynamic graph attention networks. In: Proceedings of the Twelfth ACM international conference on web search and data mining. pp. 555–563 (2019)
30. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., et al.: Graph attention networks. stat **1050**(20), 10–48550 (2017)
31. Wang, H., Zhang, F., Zhao, M., Li, W., Xie, X., Guo, M.: Multi-task feature learning for knowledge graph enhanced recommendation. In: The world wide web conference. pp. 2000–2010 (2019)
32. Wang, S., Hu, L., Wang, Y., He, X., Sheng, Q.Z., Orgun, M.A., Cao, L., Ricci, F., Yu, P.S.: Graph learning based recommender systems: A review. arXiv preprint arXiv:2105.06339 (2021)
33. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.S.: Kgat: Knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 950–958 (2019)
34. Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering. In: Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval. pp. 165–174 (2019)
35. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous graph attention network. In: The world wide web conference. pp. 2022–2032 (2019)
36. Wang, X., Liu, N., Han, H., Shi, C.: Self-supervised heterogeneous graph neural network with co-contrastive learning. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. pp. 1726–1736 (2021)
37. Wu, L., Sun, P., Fu, Y., Hong, R., Wang, X., Wang, M.: A neural influence diffusion model for social recommendation. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval. pp. 235–244 (2019)