Heterogeneous Graph Contrastive Learning for Recommendation

Mengru Chen
South China University of Technology
Guangzhou, China
cmr777qyx@gmail.com

Chao Huang*
University of Hong Kong
Hong Kong, China
chaohuang75@gmail.com

Lianghao Xia
University of Hong Kong
Hong Kong, China
aka_xia@foxmail.com

Wei Wei
University of Hong Kong
Hong Kong, China
weiweics@connect.hku.hk

Yong Xu
South China University of Technology
Guangzhou, China
yxu@scut.edu.cn

Ronghua Luo
South China University of Technology
Guangzhou, China
rhluo@scut.edu.cn

ABSTRACT

Graph Neural Networks (GNNs) have become powerful tools in modeling graph-structured data in recommender systems. However, real-life recommendation scenarios usually involve heterogeneous relationships (*e.g.*, social-aware user influence, knowledge-aware item dependency) which contains fruitful information to enhance the user preference learning. In this paper, we study the problem of heterogeneous graph-enhanced relational learning for recommendation. Recently, contrastive self-supervised learning has become successful in recommendation. In light of this, we propose a Heterogeneous Graph Contrastive Learning (HGCL), which is able to incorporate heterogeneous relational semantics into the user-item interaction modeling with contrastive learning-enhanced knowledge transfer across different views. However, the influence of heterogeneous side information on interactions may vary by users and items. To move this idea forward, we enhance our heterogeneous graph contrastive learning with meta networks to allow the personalized knowledge transformer with adaptive contrastive augmentation. The experimental results on three real-world datasets demonstrate the superiority of HGCL over state-of-the-art recommendation methods. Through ablation study, key components in HGCL method are validated to benefit the recommendation performance improvement. The source code of the model implementation is available at the link https://github.com/HKUDS/HGCL.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recommendation, Self-Supervised Learning, Contrastive Learning, Graph Neural Network, Heterogeneous Graph Representation

---

*Chao Huang is the corresponding author.

1 INTRODUCTION

In recent years, Graph Neural Networks (GNNs) have become successful in encoding relationships between users and items in recommender systems [31]. The key ideal of GNNs is to learn node (user or item) representations through the aggregation of neighboring feature information across graph propagation layers. However, many GNN-based collaborative filtering (CF) models merely focus on homogeneous interaction relationships in the generated user-item connection graphs [4, 23, 29]. In real-world recommenders, heterogeneous relational information is ubiquitous, such as social network connections between users and knowledge-aware item dependencies with semantic relatedness. In this paper, we address the challenge of incorporating heterogeneous side information into the collaborative filtering for enhancing recommender system.

Inspired by the success of GNNs in a variety of recommendation tasks, researchers attempt to design heterogeneous graph neural networks to embed rich semantics of heterogeneous relations into latent representations. However, the representation power of most existing studies are often hindered by the limitation of sparse training labels. In other words, current heterogeneous graph neural networks are label data-hungry learning models, and thus may not generate quality user/item embeddings with sparse interaction labels for model optimization of recommenders [15, 29].

Contrastive self-supervised learning, emerging as promising representation techniques for addressing data sparsity issue with data augmentation from unlabeled data itself. By integrating contrastive learning with graph neural networks, Graph Contrastive Learning (GCL) has emerged as effective solution to enhance the robustness of learned representations in the absence of sufficient observed labels [38] over graph structures. The general idea of GCL is to research the alignment between embeddings encoded from two graph contrastive representation views. In GCL-based self-supervision, the agreement between representations of positive contrastive samples will be maximized, while the distance between embeddings of

arXiv:2303.00995v1 [cs.IR] 2 Mar 2023

negative pairs will be pushed away. Motivated by this, we bring the benefits of GCL into the heterogeneous relational learning to improve recommendation performance.

However, it is non-trivial to effectively realize the heterogeneous relational learning, because the dependencies between side information and user-item interaction modeling are often not monomorphic but diverse in nature. For example, social influence among users may be different due to their personalized characteristics and diverse user-specific interaction pattern. Blindly augmenting the preference learning of users without considering their individual characteristics easily lead to suboptimal representations. In this paper, we investigate the problem of heterogeneous graph learning for recommendation by learning a contrastive augmentor. In essence, we need to solve the challenges in our designed recommender system: i) how to effectively transfer the side knowledge across different views; ii) how to perform heterogeneous relational contrastive learning with personalized augmentation.

To tackle the aforementioned challenges, we propose the principled framework, termed as Heterogeneous Graph Contrastive Learning (HGCL). Specifically, we first leverage the heterogeneous graph neural network as encoder, the rich semantics of heterogeneous relationships are preserved in the encoded embeddings. To cope with the personalized augmentation, we propose a tailored contrastive learning framework which designs a meta network to encode personalized characteristics of users and items. It allows us to perform user- and item-specific augmentation for transferring informative signals across different relational views.

The contributions of our work can be summarized as follows:

- HGCL advances the recommender system with heterogeneous graph contrastive learning, providing a general and universal framework to incorporate heterogeneous side information into recommender under a graph contrastive learning paradigm.

- HGCL solves our problem by integrating meta network with contrastive learning for adaptive augmentation to enable user-specific and item-specific knowledge transfer. It advances graph contrastive learning with customized cross-view augmentation.

- We conduct extensive experiments on real-world recommendation datasets to validate that our HGCL framework is capable of significantly improving performance over other strong baselines.

2 RELATED WORK

2.1 GNN-based Recommender Systems

In general, Graph Neural Networks (GNNs) follow the idea of message passing across different graph layers by consisting of information propagation and aggregation. Under graph neural architecture, many GNN-based recommender systems are proposed to capture various graph-structured relationships in recommendation. For example, GNNs are adopted for modeling user-item interaction graph for generating latent representations via cross-layer information propagation in NGCF [23], LR-GCCF [4] and SHT [33]. To enhance collaborative relation learning with social influence among users, social relation encoders in some existing studies are also built upon graph neural networks, such as GraphRec [6], KCGN [12] and MHCN [39]. Furthermore, graph neural networks have become effective solution to encode sequential patterns of item sequences

for time-aware recommendation, including SURGE [1] and MA-GNN [16]. In recent years, modeling multiple graph connections with GNNs (e.g., MBGCN [13] and MGNN [41]) has attracted much attention in handling more complex recommendation scenario with diverse user behaviors. In those GNN-based multi-behavior recommenders, the behavior-aware message passing is considered to reflect diverse user preference from multi-behavior data. There also exist some multimedia recommender systems (e.g., GRCN [28], DualGNN [21]) built upon graph neural networks to incorporate multi-modal information into recommendation.

2.2 Contrastive Learning for Recommendation

Recently, the contrastive self-supervised learning has been noticed by researchers. It is because the generated self-supervision signals can be used to enrich user representation learning. In recommender systems, contrastive learning can be a powerful tool to incorporate self-supervision signals for data augmentation with the alignment between contrastive representation views. For example, many studies aim to address the data sparsity issue in recommenders by proposing various graph augmentation schemes for embedding contrasting, e.g., SGL [29], HCCF [32] and NCL [14]. In particular, random node/edge dropout operations are adopted for generate graph contrastive learning views in SGL [29]. In HCCF [32], local-global contrastive learning is designed for self-supervised augmentation based on parameterized hypergraph structures. In those contrastive graph CF models, the embedding uniformity can be improved based on InfoNCE-based contrasting. There also exist some studies leveraging contrastive learning in knowledge graph representation in recommender systems, such as KGCL [37] and KGIC [43]. In addition, contrastive learning has been used in various recommendation scenarios, including sequential recommendation [26], multi-behavior recommendation [27], and multi-interest recommendation [40]. In this work, a novel heterogeneous graph contrastive learning paradigm is proposed to fill the gap in recommender system by capturing heterogeneous relationships in recommendation with contrastive learning.

2.3 Heterogeneous Graph Learning

Heterogeneous graphs is ubiquitous in real-life applications with various types of nodes and connections. Representation learning over heterogeneous graphs aims to encode node embeddings in which the rich semantics with relation heterogeneity can be well preserved [35]. To achieve this goal, heterogeneous graph neural networks become the promising techniques to provide state-of-the-art representation results. For example, HAN [24] enhances the graph attention network with the capability of dealing with heterogeneous types of nodes and relations based on meta-path construction. Motivated by the transformer framework, HGT [11] designs a graph transformer network to enable the heterogeneous message passing using self-attention to calculate the propagation weights between nodes. In addition, both intra- and inter-metapath aggregation are considered in MAGNN [7] to fuse information from different meta-paths over heterogeneous graphs. In HGIB [36], information bottleneck is extended to heterogeneous graph learning with self-supervision among homogeneous graphs. Towards this

research line, this paper tackles an important but unexplored task of heterogenous graph contrastive learning recommendation.

3 METHODOLOGY

In this section, we elaborate the model design of our proposed HGCL framework, which enhances representation learning on heterogenous graph for recommendation with cross-view contrastive learning. The overall framework of HGCL is illustrated in Figure 1.

3.1 Preliminaries

Relations in real-life recommender systems are often heterogeneous to contain diverse semantic information from users and items. We represent the user-item interaction data with the graph $\mathcal{G}_{ui} = \{\mathcal{V}_u, \mathcal{V}_i, \mathcal{E}_{ui}\}$, where $\mathcal{V}_u$ and $\mathcal{V}_i$ denote the sets of users and items, respectively. In graph $\mathcal{G}_{ui}$, if user u has adopted item i, then there exists an edge between u and i $((u, i) \in \mathcal{E}_{ui})$. To represent social relationships among users, graph $\mathcal{G}_{uu} = \{\mathcal{V}_u, \mathcal{E}_{uu}\}$ is defined to include user-wise social connections with the edge set $\mathcal{E}_{uu}$. To incorporate item-wise relations, we define the item graph $\mathcal{G}_{ii} = \{\mathcal{V}_i, \mathcal{E}_{ii}\}$ to connect dependent items with external knowledge (e.g., item category). For these defined graphs, we define three adjacent matrices $\mathbf{A}_{ui} \in \mathbb{R}^{m \times n}, \mathbf{A}_{uu} \in \mathbb{R}^{m \times m}$ and $\mathbf{A}_{ii} \in \mathbb{R}^{n \times n}$, corresponding to graph $\mathcal{G}_{ui}, \mathcal{G}_{uu}$ and $\mathcal{G}_{ii}$, respectively. Here, m and n denotes the number of users and items, respectively. The objective of this work is to predict unobserved interactions between users and items given the graphs with relation heterogeneity.

3.2 Heterogeneous Graph Relation Learning

3.2.1 **Relation-Aware Embedding Initialization.** To encode the heterogeneous collaborative relations with the modeling of high-order connectivity, we employ heterogeneous graph neural networks to learn embeddings from the user-item graph $\mathcal{G}_{ui}$, user-user graph $\mathcal{G}_{uu}$, and item-item graph $\mathcal{G}_{ii}$. To begin with, we assign id-corresponding embeddings $\mathbf{e}_u, \mathbf{e}_i \in \mathbb{R}^d$ initialized by xavier initializer [8], where d denotes the hidden dimensionality. The node-specific embeddings form the initial embedding matrices $\mathbf{E}_u^0 \in \mathbb{R}^{m \times d}$ and $\mathbf{E}_i^0 \in \mathbb{R}^{n \times d}$. The initial embeddings are fed into different graph encoders for user-item domain, user-user domain, and item-item domain. To highlight the differences in interactive patterns between the three relation types, we train a self-gating module [39] to derive the relation-aware embeddings for user-wise social connections and item-wise semantic relations from the common initial embedding space, which are showned as follows:

$$\mathbf{E}_{uu}^0 = \mathbf{E}_u^0 \odot \sigma(\mathbf{E}_u^0 \mathbf{W}_g + \mathbf{b}_g); \quad \mathbf{E}_{ii}^0 = \mathbf{E}_i^0 \odot \sigma(\mathbf{E}_i^0 \mathbf{W}_g + \mathbf{b}_g) \quad (1)$$

where $\mathbf{E}_{uu}^0 \in \mathbb{R}^{m \times d}$ and $\mathbf{E}_{ii}^0 \in \mathbb{R}^{n \times d}$ are the embeddings for the homogeneous graphs $\mathcal{G}_{uu}$ and $\mathcal{G}_{ii}$ for user-wise and item-wise relations, respectively. $\sigma(\cdot)$ denotes the sigmoid activation function. $\odot$ denotes element-wise multiplication operation. $\mathbf{W}_g \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_g \in \mathbb{R}^{d \times 1}$ are the transformation and bias parameters. Through the self-gating mechanism with multiplicative skip-connection [5], embeddings $\mathbf{E}_{uu}^0, \mathbf{E}_{ii}^0$ not only share common semantic with initial embeddings $\mathbf{E}_u^0, \mathbf{E}_i^0$ for user-item interactions, but also gain the flexibility to characterize the user-user and item-item relations.

3.2.2 **Heterogeneous Message Propagation.** Among the above initial embedding matrices, $\mathbf{E}_u^0, \mathbf{E}_i^0$ are used as input for the user-item view, $\mathbf{E}_{uu}^0$ and $\mathbf{E}_{ii}^0$ are used as input for the user-user view and the item-item view, respectively. We first apply a graph convolutional neural network as the encoder for three views of graph structures. Without loss of generality, we elaborate the modeling for user-item relation graph as an example. Specifically, given the user-item interaction graph $\mathcal{G}_{ui}$, our HGCL iteratively refine the user and item embeddings with the message propagation as follows:

$$\mathbf{e}_u^{l+1} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} \mathbf{e}_i^l; \quad \mathbf{e}_i^{l+1} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^l \quad (2)$$

where $\mathcal{N}_u$ and $\mathcal{N}_i$ denote the neighbor set of target nodes u and i, respectively. $\mathbf{e}_u^l, \mathbf{e}_i^l \in \mathbb{R}^d$ denotes the embedding vectors of user u and item i in the l-th iteration. $\mathbf{e}_u^0, \mathbf{e}_i^0$ are row vectors of the embedding matrices $\mathbf{E}_u^0, \mathbf{E}_i^0$, respectively. Inspired by the effectiveness and efficiency of lightweight GCN [9] in CF recommendation, our relation-aware message passing paradigm is configured without transformation and non-linear activation. Analogously, the embeddings $\mathbf{E}_{uu}^l$ for user-user graph and the embeddings $\mathbf{E}_{ii}^l$ for item-item graph are refined iteratively following the same GCN schema.

3.2.3 **Heterogeneous Information Aggregation.** Inspired by the *soft meta-path* design in [11], the information in each iteration is aggregated from heterogeneous relations. Through multiple iterations of heterogeneous message propagation, the high-order embeddings preserve heterogeneous semantics with multi-hop connections. In particular, the embddings of users and items are updated through the following defined heterogeneous fusion procedure:

$$\widehat{\mathbf{E}}_u^{l+1} = f(\mathbf{E}_u^{l+1}, \mathbf{E}_{uu}^{l+1}); \quad \widehat{\mathbf{E}}_i^{l+1} = f(\mathbf{E}_i^{l+1}, \mathbf{E}_{ii}^{l+1}) \quad (3)$$

where the refined embeddings in the $l + 1$ iteration $\widehat{\mathbf{E}}_u^{l+1} \in \mathbb{R}^{m \times d}$, $\widehat{\mathbf{E}}_i^{l+1} \in \mathbb{R}^{n \times d}$ integrate heterogeneous semantics and become the input for the next layer. f denotes the heterogeneous information fusion function. Here, to reduce the model complexity, we use element-wise mean pooling as the fusion function $f(\cdot)$.

To further aggregate heterogeneous information with encoded layer-specific representations ($1 \leq l \leq L$), we generate the overall embeddings of users and items as follows:

$$\mathbf{E}_u = \mathbf{E}_u^0 + \sum_{l=1}^{L} \frac{\mathbf{E}_u^l}{||\mathbf{E}_u^l||}; \quad \mathbf{E}_i = \mathbf{E}_i^0 + \sum_{l=1}^{L} \frac{\mathbf{E}_i^l}{||\mathbf{E}_i^l||} \quad (4)$$

where L denotes the maximum number of GCN iterations. The output of each GCN layer are normalized. We add the initial embeddings $\mathbf{E}_u^0, \mathbf{E}_i^0$ using skip connections. The above presented formulas indicate the layer-specific representation aggregation for the user-item interaction view. The embeddings of user-user social view (i.e. $\mathbf{E}_{uu}$) and the item-item dependency view (i.e. $\mathbf{E}_{ii}$) are obtained through multi-order information aggregation in an analogous way.

3.3 Cross-View Meta Network

Our HGCL aims to enhance the collaborative filtering by incorporating the heterogeneous relational knowledge from both user social connections and item external dependence. However, in real-life user modeling scenario, the influence of user and item side
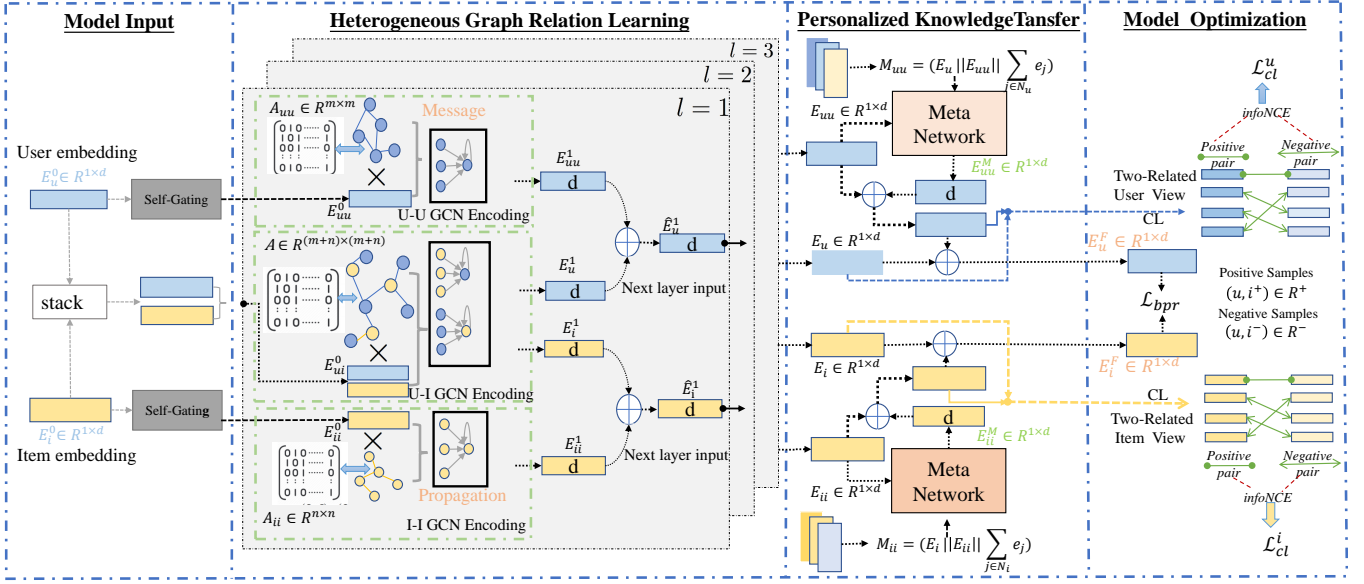
Figure 1: The model flow of the proposed HGCL framework. HGCL includes three key components: (1) Heterogeneous graph representation extraction and fusion by heterogeneous graph neural network on user-user graph, user-item graph and item-item graph. (2) Meta network for personalized cross-view dependencies modeling between the auxiliary views and the interaction view. (3) Jointly parameter optimization with adaptive contrastive learning between the heterogeneous relational views.

information over the user-item interaction patterns may be different among users. For example, some users are more likely to be influenced by the recommendations from their social friends, while others often adopt items based on their own preference. Therefore, it is necessary to perform personalized knowledge transfer from side information to guide the learning of user-specific preference. Towards this end, we design a cross-view meta network to enable the customized knowledge distillation from both user and item side.

3.3.1 **Meta Knowledge Extraction.** To generate personalized mapping from the auxiliary views (user and item side information) to the encoding of user-item interaction for each user and each item, we first extract meta knowledge to preserve important features of users and items *w.r.t* both the auxiliary views and interaction view. Specifically, the distilled meta knowledge for the user-user relation view and the item-item relation view is obtained as follows:

$$\mathbf{M}_{uu} = \mathbf{E}_u||\mathbf{E}_{uu}|| \sum_{i \in \mathcal{N}_u} \mathbf{e}_i; \quad \mathbf{M}_{ii} = \mathbf{E}_i||\mathbf{E}_{ii}|| \sum_{u \in \mathcal{N}_i} \mathbf{e}_u \quad (5)$$

where $\mathbf{M}_{uu} \in \mathbb{R}^{m \times 3d}, \mathbf{M}_{ii} \in \mathbb{R}^{n \times 3d}$ represent the meta knowledge that encodes the context information to generate personalized knowledge transfer functions for user and item side knowledge, respectively. Motivated by [34], the meta knowledge contains the node representation $\mathbf{E}_{uu}, \mathbf{E}_{ii}$ of the source domains (*i.e.* the user-user and item-item relation view) as well as the embeddings of the target user-item interaction view $\mathbf{E}_u, \mathbf{E}_i$. In addition, we incorporate the neighborhood information into the meta knowledge. Specifically, the embeddings of the auxiliary domains characterize the users' social influence and item semantic relatedness. The

embeddings of the user-item view captures the item-related interactive patterns of users. The additional neighborhood information explicitly enhances the modeling of direct graph connections. By collectively considering the three dimensions of information, the meta knowledge is able to well-reflect the important contextual signals for personalized cross-view knowledge transferring.

3.3.2 **Personalized Cross-View Knowledge Transfer.** In our HGCL, the extracted meta knowledge is utilized to generate a parameterized knowledge transfer network with customized transformation matrices. The proposed meta neural network is

$$\begin{cases} f_{mlp}^1(\mathbf{M}_{uu}) \rightarrow \mathbf{W}_{uu}^{M1} \\ f_{mlp}^2(\mathbf{M}_{uu}) \rightarrow \mathbf{W}_{uu}^{M2} \end{cases} \quad (6)$$

where f_{mlp}^1, f_{mlp}^1 are meta knowledge learner consisting of two fully-connected layers with PReLU activation function. The functions take the meta knowledge $\mathbf{M}_{uu}$ as input, and output the customized transformation matrices $\mathbf{W}_{uu}^{M1} \in \mathbb{R}^{m \times d \times k}, \mathbf{W}_{uu}^{M2} \in \mathbb{R}^{m \times k \times d}$. Both two parameter tensors contain m matrices for each of the m users. The customized transformations are generated according to the unique characteristics of the corresponding users and items to realize the personalized knowledge transfer. The two sets of matrices restrict the rank of the transformation to $k < d$, which not only reduces the number of trainable parameters of meta knowledge learnder and enhance the model stability. Inspired by the personalized bridge function in [42], we leverage the generated parameter matrices and a non-linear mapping function to build our customized

transfer network as follows:

$$\mathbf{E}_{uu}^M = \sigma(\mathbf{W}_{uu}^{M1}\mathbf{W}_{uu}^{M2}\mathbf{E}_{uu}) \tag{7}$$

where $\sigma(\cdot)$ denotes the PReLU activate function. $\mathbf{E}_{uu}^M \in \mathbb{R}^{m\times d}$ contains the embeddings transformed by the customized mapping function for the user-user social view. Then the customized embeddings are utilized to enhance the user embeddings encoded from the user-item interactions. The fusion process for users is conducted by the following weighted summation:

$$\mathbf{E}_u^F = \alpha_u * \mathbf{E}_u + (1 - \alpha_u) * (\mathbf{E}_{uu} + \mathbf{E}_{uu}^M); \tag{8}$$

where $\alpha_u \in \mathbb{R}$ denotes the hyperparameter which controls the weight between the user-item interaction view embedding and the user-user social view embedding. Here the original embeddings of user-user relation view is also utilized for better optimization. $\mathbf{E}_u^F \in \mathbb{R}^{m\times d}$ represent the final embeddings used for the main task of recommendation. The foregoing process elaborates the calculation for cross-view user embedding customization. The cross-view item embeddings $\mathbf{E}_{ii}^M, \mathbf{E}_i^F$ can be generated in a similar way.

3.4 Heterogeneous Relational Contrastive Learning for Augmentation

3.4.1 **Cross-View Contrastive Learning.** To further enhance the representation learning of our HGCL framework with more supervision signals to mitigate the data sparsity issue, we design the cross-view contrastive learning paradigm to enhance the robustness of the heterogeneous relational learning with self-augmentation. Concretely, the embeddings of the two auxiliary views (*i.e.* $\mathbf{E}_{uu}^M$ and $\mathbf{E}_{ii}^M$) are aligned with the embeddings of the user-item interaction view (*i.e.* $\mathbf{E}_u$ and $\mathbf{E}_i$). With this design, the embeddings of the auxiliary views serve as effective regularization to influence the user-item interaction modeling with the self-supervised signals.

To capture the diverse user preference by considering the personalized cross-view knowledge transfer, we integrate the personalized cross-view knowledge transfer with the contrastive learning in our recommender system. In particular, the cross-view embedding alignment is conducted in an adaptive way between different representation views. The auxiliary-view-specific embeddings $\mathbf{E}_{uu}, \mathbf{E}_{ii}$ are processed by the personalized mapping functions generated by the meta network, to yield the personalized auxiliary embeddings $\mathbf{E}_{uu}^M, \mathbf{E}_{ii}^M$. The meta network is trained to filter noisy features in the auxiliary views to match the user-item interaction view.

3.4.2 **InfoNCE-based Contrastive Loss.** With the help of our heterogeneous graph relation learning and cross-view meta networks, we obtain two sets of embeddings for both users and items, *i.e.* $\mathbf{E}_{uu}^M, \mathbf{E}_u$ for users, and $\mathbf{E}_{ii}^M, \mathbf{E}_i$ for items. The embeddings are obtained via encoding the user-item interaction data, and the user/item-side auxiliary knowledge. Inspired by the success of recent contrastive self-supervised learning in recommendation [29, 32], we propose to empower the user/item representation learning of our HGCL method with the InfoNCE-based contrastive learning loss between two representation views as follows:

$$\mathcal{L}_{cl}^u = \sum_{u\in\mathcal{V}_u} -\log\frac{\exp\left(s(\mathbf{e}_{uu}^M + \mathbf{e}_{uu}, \mathbf{e}_u)/\tau\right)}{\sum_{u'\in\mathcal{V}_u}\exp\left(s(\mathbf{e}_{uu}^M + \mathbf{e}_{uu}, \mathbf{e}_u')/\tau\right)} \tag{9}$$

where $\mathbf{e}_{uu}^M \in \mathbb{R}^d$, $\mathbf{e}_u \in \mathbb{R}^d$ are the embedding vectors from the matrices $\mathbf{E}_{uu}^M$ and $\mathbf{E}_u$, respectively. $s(\cdot)$ denotes the similarity function, which can be inner product or cosine similarity. Here we use cosine similarity as our $s(\cdot)$. τ represents the temperature coefficient, which is capable of automatically identifying difficult negative samples. u' indicates negative samples with different indices. Analogously, we can obtain the InfoNCE loss $\mathcal{L}_{cl}^i$ of items aspect. Finally, the total contrastive loss is $\mathcal{L}_{cl} = \alpha_1 * \mathcal{L}_{cl}^u + \alpha_2 * \mathcal{L}_{cl}^i$, where α_1 and α_2 denote two hyperparameters for weight tuning.

3.5 Optimization Objectives of HGCL

With the fused embeddings $\mathbf{E}_u^F, \mathbf{E}_i^F$, our HGCL forecast the likelihood of user u interacting with item i via dot-product: $\hat{y}_{u,i} = \mathbf{e}_u^{F\top}\mathbf{e}_i^F$, where $\mathbf{e}_u^F$ and $\mathbf{e}_i^F$ denote the final embedding vectors of user u and item i from the fused embedding matrices. $\hat{y}_{u,i} \in \mathbb{R}$ denotes the score that indicates the likelihood of user u interacting with item i. Larger $\hat{y}_{u,i}$ reflects larger probability of interaction. To optimize our HGCL with the recommendation task, we follow recent works and adopt the Bayesian Personalized Ranking (BPR) [17] pair-wise loss function. Specifically, each training sample is configured with a user u, a positive item i^+ that the user has interacted with, and a negative item i^- that the user has not interacted with. For each training sample, we maximize the prediction score as follows:

$$\mathcal{L}_{bpr} = \sum_{(u,i^+,i^-)\in O} -\ln(\text{sigmoid}(\hat{y}_{u,i^+} - \hat{y}_{u,i^-})) + \lambda||\Theta||^2 \tag{10}$$

where $\ln(\cdot)$ and $\text{sigmoid}(\cdot)$ denote the logarithm function and the sigmoid function, respectively. λ denotes a hyperparameter to determine the weight of the regularization term. Combining the BPR loss function with the augmented cross-view contrastive learning loss, the overall training loss is presented as follows:

$$\mathcal{L} = \mathcal{L}_{bpr} + \beta * \mathcal{L}_{cl} \tag{11}$$

3.6 Model Complexity Analysis

We give detailed analysis on the time complexity of our HGCL model to measure the efficiency of our method. The heterogeneous GNN module of HGCL employs a lightweight network structure, which takes $O((|\mathcal{E}_{ui}| + |\mathcal{E}_{uu}| + |\mathcal{E}_{ii}|) \times d\times\text{L})$ time. In the cross-view meta network, the highest computational cost comes from the meta network for personalized mapping function generation, which takes $O((m + n) \times d^2 \times k)$ time. In the heterogeneous relational contrastive learning component, $O(b\times(m+n)\times d)$ time complexity is needed in each batch (batch size d) to calculate the InfoNCE loss across the heterogeneous relational views. Overall, the above discussed first and third components are identical to the complexity of state-of-the-art self-supervised GNN recommendation methods (*e.g.*, SGL [29]). The second module takes the complexity which is close to the complexity of a vanilla GNN as k is typically small.

4 EVALUATION

In this section, we perform model evaluation to investigate the effectiveness of our HGCL and baseline methods. We also analyze the impact of key modules and model robustness. Our experiments are designed to address the following research questions:

Table 1: Performance comparison of all methods on different datasets in terms of *NDCG* and *HR*.

Data	Metric	SAMN	DGRec	ETANN	NGCF	KGAT	MKR	GraphRec	DANSER	HERec	MCRec	HAN	HeCo	HGT	MHCN	SMIN	**HGCL**
Ciao	H@10	0.6576	0.6653	0.6738	0.6945	0.6601	0.6793	0.6825	0.6730	0.6800	0.6772	0.6589	0.6867	0.6939	0.7053	0.7108	**0.7376**
	N@10	0.4561	0.4953	0.4665	0.4894	0.4512	0.4589	0.4730	0.4521	0.4712	0.4708	0.4469	0.4867	0.4869	0.4928	0.5012	**0.5261**
Epinions	H@10	0.7592	0.7603	0.7650	0.7984	0.7510	0.7647	0.7723	0.7714	0.7642	0.7630	0.7505	0.7998	0.8150	0.8201	0.8179	**0.8367**
	N@10	0.5614	0.5668	0.5663	0.5945	0.5578	0.5669	0.5751	0.5741	0.5495	0.5326	0.5275	0.5910	0.6126	0.6158	0.6137	**0.6413**
Yelp	H@10	0.7910	0.7950	0.8031	0.8265	0.7881	0.8005	0.8098	0.8077	0.7928	0.7869	0.7731	0.8359	0.8364	0.8344	0.8478	**0.8712**
	N@10	0.5516	0.5593	0.5560	0.5854	0.5501	0.5635	0.5679	0.5692	0.5612	0.5590	0.5604	0.5847	0.5883	0.5799	0.5993	**0.6310**

Table 2: Statistics of experimented datasets

Dataset	User #	Item #	Interaction #	Sparsity
Ciao	6776	101415	265308	99.9614%
Epinions	15210	233929	630391	99.9823%
Yelp	161305	114852	957923	99.9948%

- **RQ1**: How does HGCL perform compared with existing methods?
- **RQ2**: Is it beneficial to incorporate key components in our HGCL to boost the recommendation performance?
- **RQ3**: How doe HGCL perform in different environments with varying sparsity degrees of user interaction data?
- **RQ4**: How does key hyperparameters affect model performance?

4.1 Experimental Settings

4.1.1 **Datasets.** In our experiments, our HGCL framework is evaluated on three real-world datasets from online platforms. We present the data statistics in Table 2 and present the details of each dataset as followed. **Ciao and Epinions**. They are two benchmark recommendation datasets collected from online review systems to contain user rating behaviors over different items. The heterogeneous relations are generated from the contained user and item side information, such as user trust relationships and item categorical information. **Yelp**. This dataset contains heterogeneous relations (*e.g.*, user social relations, venue rating behaviors, business attributes) in the recommendation scenario of local businesses on Yelp platform.

4.1.2 **Baselines.** To evaluate the validity of our proposed method, we compare HGCL with various systems for comprehensive performance comparison. The baseline details are described as below.

- **SAMN** [2]: This model designs attention-based memory network to consider the difference of social influence among users for improving the user-item interaction modeling.
- **DGRec** [19]: This approach utilizes recurrent neural network to model dynamic interests of users and graph attention network to model social influence for recommendation.
- **ETANN** [3]: It designs an adaptive transfer scheme from the social domain to the encoding process of user-item interaction patterns by considering user-user relationships.
- **NGCF** [23]: We incorporate the social information among users into the representative GNN-based collaborative filtering model. The message passing is built based on graph convolutions.
- **KGAT** [22]: In this baseline, the item knowledge-based relationships are incorporated into the graph attention mechanism for

enhancing recommender system.

- **MKR** [20]: It utilizes knowledge graph as the side information to assist the recommendation with multi-task learning framework. Different tasks are associated with cross and compress units.
- **GraphRec** [6]: This method jointly models the user-user social graph and user-item interaction graph to reflect the relation heterogeneity in recommendation.
- **DANSER** [30]: This recommender system learns two-fold of social effects with user-specific and dynamic attentive weights estimated via contextual multi-armed bandit.
- **HERec** [18]: It aims to encode heterogeneous information in recommendation based on meta-path-based random walk.
- **MCRec** [10]: Co-attention mechanism is proposed to capture the heterogeneous relationships in recommender system.
- **HAN** [24]: We apply this representative heterogeneous graph neural network to generate user and item representations via meta-path-based attention encoder.
- **HGT** [11]: It introduces heterogeneous mutual attention for message passing scheme to refine user/item embeddings along with diverse relations in the heterogeneous graph structures.
- **HeCo** [25]: It is a self-supervised method which integrates contrastive learning with heterogeneous GNNs to consider local and high-order graph structures. Embeddings encoded with different meta-path-based connections are used for contrasting.
- **SMIN** [15]: It is a self-supervised social recommender system which incorporates auxiliary graph learning task into the main task to improve the recommendation performance.
- **MHCN** [39]: In this recommender, a multi-channel hypergraph convolutional network is designed to consider global relationships among users based on motifs.

4.1.3 **Hyperparameter Settings.** Our HGCL model is implemented using PyTorch. The model is optimized with Adam for parameter learning. In the model implementation, the batch size and learning rate is searched from {1024, 2048, 4096, 8192} and {4e-2, 4.5e-2, 5e-2, 5.5e-2, 6e-2}, respectively. The embedding size is tuned from the range of {8, 16, 32, 64, 128}. The number of graph neural network layers is selected from {1, 2, 3}. Additionally, the coefficient β of contrastive loss is selected from {0.2, 0.25, 0.3, 0.35, 0.55, 0.6, 0.65}. The dimension of low rank matrix decomposition of meta knowledge extraction is chosen from {1, 2, 3, 4, 5}.

In our evaluation settings, one positive (interacted) item and 99 negative (non-interacted) items are sampled for each user for performance evaluation. To measure the recommendation accuracy
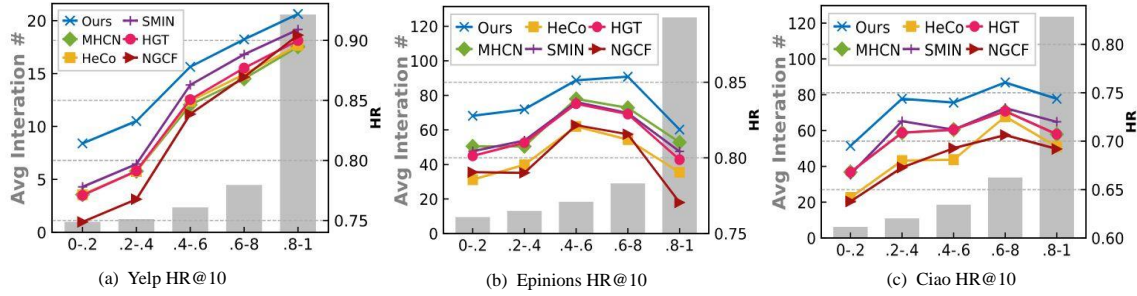
Figure 2: Performance comparison with respect to different data sparsity degrees on three datasets.

Table 3: Ablation study on key components of HGCL

Data	Ciao		Epinions		Yelp	
Metric	HR	NDCG	HR	NDCG	HR	NDCG
w/o-cl	0.7124	0.5015	0.8176	0.6166	0.8471	0.6030
w/o-meta	0.7215	0.5135	0.8247	0.6282	0.8585	0.6218
w/o-ii	0.7116	0.5055	0.8245	0.6317	0.8573	0.6188
w/o-uu	0.7149	0.5047	0.8285	0.6266	0.8533	0.6208
HGCL	**0.7376**	**0.5261**	**0.8367**	**0.6413**	**0.8712**	**0.6310**

of different methods, two widely-adopted metrics HR(Hit Ratio) and NDCG (Normalized Discounted Cumulative Gain) are used.

4.2 Performance Comparison (RQ1)

Table 1 reports the performance of all compared methods on different datasets for item recommendation. From the evaluation results, we summarize the following key observations:

- Our HGCL consistently achieves the significant performance improvement compared with state-of-the-arts. We attribute these improvements to the design of heterogeneous graph contrastive learning: (1) HGCL allows recommender system to perform effective knowledge transfer among heterogeneous relationships to help model user preference; (2) the adaptive contrastive learning has great ability to improve recommendation performance with self-supervision signals between heterogeneous relation views.

- Heterogeneous graph neural network-based methods (*e.g.*, HeCo, HGT, SMIN) often offer better performance than other alternative approaches (*e.g.*, SAMN, KGAT, DANSER), which justifies the effectiveness of incorporating heterogeneous relational knowledge of social influence and item semantic relatedness from user and item side into the recommender system.

- As can be seen, the observed superior performance of MHCN and SMIN indicates the rationality of augmenting user-item interaction encoding with self-supervised learning technique. The performance gap between our HGCL and those self-supervised learning-enhanced recommenders validates that adaptive self-supervised signal distillation indeed boosts the performance with contrastive personalized knowledge transfer.

4.3 Ablation Study (RQ2)

We conduct ablation study to validate that the consideration of customized contrastive learning heterogeneous relationships is essential and benefit the performance, as elaborated below:
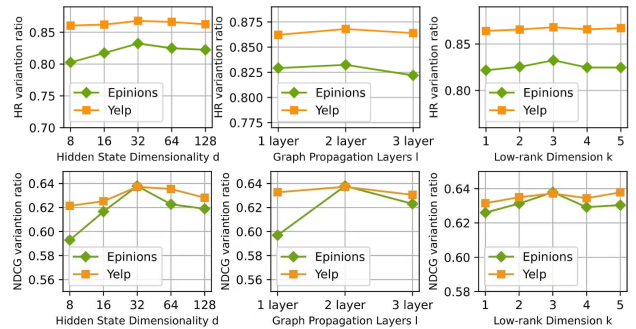

Figure 3: Hyperparameter study of the HGCL.

- ***w/o-meta***: We do not include the meta network in HGCL to allow the personalized knowledge transfer in our developed contrastive learning augmentation across heterogeneous relational views.

- ***w/o-cl***: We disable the contrastive learning in our model to capture the cross-view dependency between the auxiliary information and user-item interaction modeling.

- ***w/o-ii***: In this variant, we do not include the item-item graph $\mathcal{G}_{ii}$ to capture the knowledge-aware dependency among items for guiding the learning process of user preference.

- ***w/o-uu***: In this variant, we do not include the user-user graph $\mathcal{G}_{uu}$ to consider the social influence among users to help encode the user-item interaction patterns.

The recommendation performance of HGCL framework and compared variants are presented in Table 3. In all cases, the performance of HGCL is superior to w/o-cl, reflects the rationalities of our heterogeneous graph contrastive learning for effective augmentation with cross-view knowledge transfer. w/o-meta performs worse than HGCL on different datasets. This result is consistent with our assumption that user/item-specific customized knowledge transfer is helpful to learn user representations. HGCL achieves consistent gain over w/o-ii and w/o-uu, which implies the necessity of considering heterogeneous side information into the recommender system to guide the encoding of user preference.

4.4 Performance varying Data Sparsity

In this section, we evaluate the performance of different methods when varying the data sparsity degrees of user interaction data. We divide the set of users into five groups to represent diverse
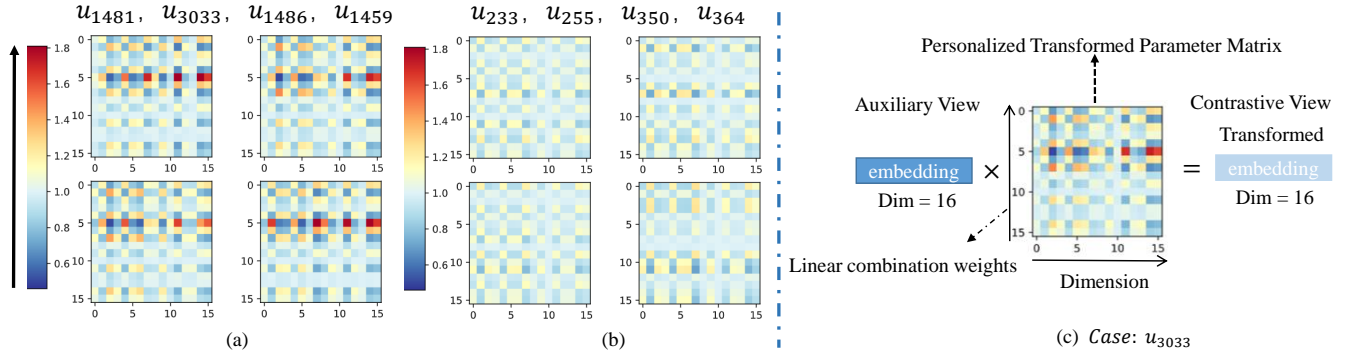
User Cases from Ciao Dataset

Figure 4: Case study on Ciao dataset to visualize the learned contrastive transformation matrices sampled from different users to reflect the diverse social influence. (a): Four users who are more likely to be influenced by their social relations; (b): Four users who are less likely to be influenced by their social relations. (c): The embeddings generated from auxiliary view will be transformed for representation contrasting for self-supervision augmentation of user-item interaction modeling.

user active degrees. The performance comparison results between HGCL and several baselines are shown in Figure 2. The recommendation accuracy of each method is presented in the right side of y-axis with lines. The left side y-axis represents the number of average number of interactions in each user group with bars. It is obvious to see the superior performance of our method under different sparsity environments. The improvements of HGCL may come from the contrastive learning-enhanced cross-view knowledge transfer, because it can effectively capture the user-specific social influence and item-specific semantic relatedness. Therefore, through the conducted experiments, HGCL is able to maintain a decent performance even with sparse user-item interactions.

4.5 Hyperparameter Analysis

We further perform parameter sensitivity analysis to show the impact of hidden state dimensionality, the number of graph propagation layers, and low-rank dimension. The results are shown in Figure 3. From the results, we make the following conclusions.

- **Hidden State Dimensionality**. The hidden state dimensionality d is selected from 8 to 128. We can notice that the model performance firstly increases and then reaches saturation when $d = 32$. Hence, properly enlarging the embedding dimension size can boost the recommendation performance, but not always being performance gain due to model overfitting.
- **The Number of Graph Propagation Layers**. In graph neural architecture, the number of propagation layers is searched from 1 to 3. The curves depicts that the model achieves the better performance by stacking two layers. This suggests that more layers could capture the high-order neighbors and semantic information. However, deeper GNN architecture can lead to the model over-smoothing and induce noise to the feature representation.
- **Low-rank Decomposition Dimension**. We can observe that the parameter study on the low-rank decomposition dimension k indicates that the best performance is obtained with $k = 3$.

Smaller value of k may not be sufficient to learn the complex transformation information.

4.6 Qualitative Evaluation

In our evaluation, we perform case studies on Ciao dataset to visualize the learned personalized contrastive transformation matrix ($\mathbb{R}^{16 \times 16}$) to reflect the diverse influence between the auxiliary view (*e.g.*, social relationships) and the user-item interaction view. In Figure 4, we sample four users who are more (*e.g.*, u_{1481}, u_3033)/less (*e.g.*, u_{233}, u_{255}) likely to be influenced by social relationships when adopting items. The corresponding personalized contrastive transformation matrices of different users are visualized to capture diverse knowledge transfer between the social view and interaction view. We can observe that larger values in the learned contrastive transformation matrix indicate larger social influence for this user. With the integration of meta network and contrastive learning, the adaptive contrastive data augmentation can be realized based on the personalized characteristics of users.

5 CONCLUSION

In this paper, we study the problem of graph representation learning for recommendation with the consideration of heterogeneous relations. To solve this problem, a novel heterogeneous graph contrastive learning model (HGCL) is proposed to transfer knowledge from side information to the user-item interaction modeling in an adaptive way. In our HGCL, we propose to identify the informative heterogeneous relations to augment collaborative filtering paradigm. Our experiments on real-world datasets validate that our HGCL outperforms state-of-the-arts by a large margin. In-depth analysis validates the robustness of our model in alleviating data sparsity. One interesting direction for future work is to explore and disentangle the real interest and conformity, by incorporating heterogeneous relationships in recommender systems to alleviate the popularity bias from noisy interaction data of users. Furthermore, in future work, it is also interesting to explore confounding effects for heterogeneous relational learning in recommender systems.

REFERENCES

[1] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential recommendation with graph neural networks. In *SIGIR*. 378–387.

[2] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2019. Social attentional memory network: Modeling aspect-and friend-level differences in recommendation. In *WSDM*. 177–185.

[3] Chong Chen, Min Zhang, Chenyang Wang, Weizhi Ma, Minming Li, Yiqun Liu, and Shaoping Ma. 2019. An efficient adaptive transfer neural network for social-aware recommendation. In *SIGIR*. 225–234.

[4] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *AAAI*, Vol. 34. 27–34.

[5] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *International conference on machine learning*. PMLR, 933–941.

[6] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *WWW*. 417–426.

[7] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *WWW*. 2331–2341.

[8] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*. JMLR Workshop and Conference Proceedings, 249–256.

[9] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.

[10] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *KDD*. 1531–1540.

[11] Ziniu Hu, Yuxiao Dong, Kuansan Wang, et al. 2020. Heterogeneous graph transformer. In *WWW*. 2704–2710.

[12] Chao Huang, Huance Xu, Yong Xu, Peng Dai, Lianghao Xia, Mengyin Lu, Liefeng Bo, Hao Xing, Xiaoping Lai, and Yanfang Ye. 2021. Knowledge-aware coupled graph neural network for social recommendation. In *AAAI*, Vol. 35. 4115–4122.

[13] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Multi-behavior recommendation with graph convolutional networks. In *SIGIR*. 659–668.

[14] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning. In *WWW*. 2320–2329.

[15] Xiaoling Long, Chao Huang, Yong Xu, Huance Xu, Peng Dai, Lianghao Xia, and Liefeng Bo. 2021. Social Recommendation with Self-Supervised Metagraph Informax Network. In *CIKM*. 1160–1169.

[16] Chen Ma, Liheng Ma, Yingxue Zhang, Jianing Sun, Xue Liu, and Mark Coates. 2020. Memory augmented graph neural networks for sequential recommendation. In *AAAI*, Vol. 34. 5045–5052.

[17] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[18] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. 2018. Heterogeneous information network embedding for recommendation. *TKDE* 31, 2 (2018), 357–370.

[19] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based social recommendation via dynamic graph attention networks. In *WSDM*. 555–563.

[20] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-task feature learning for knowledge graph enhanced recommendation. In *WWW*. 2000–2010.

[21] Qifan Wang, Yinwei Wei, Jianhua Yin, Jianlong Wu, Xuemeng Song, and Liqiang Nie. 2021. DualGNN: Dual Graph Neural Network for Multimedia Recommendation. *Transactions on Multimedia* (2021).

[22] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *KDD*. 950–958.

[23] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR*.

[24] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *WWW*. 2022–2032.

[25] Xiao Wang, Nian Liu, Hui Han, and Chuan Shi. 2021. Self-supervised heterogeneous graph neural network with co-contrastive learning. In *KDD*. 1726–1736.

[26] Ziyang Wang, Huoyu Liu, Wei Wei, Yue Hu, Xian-Ling Mao, Shaojian He, Rui Fang, and Dangyang Chen. 2022. Multi-level Contrastive Learning Framework for Sequential Recommendation. In *CIKM*. 2098–2107.

[27] Wei Wei, Chao Huang, Lianghao Xia, Yong Xu, Jiashu Zhao, and Dawei Yin. 2022. Contrastive meta learning with behavior multiplicity for recommendation. In *WSDM*. 1120–1128.

[28] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2020. Graph-refined convolutional network for multimedia recommendation with implicit feedback. In *MM*. 3541–3549.

[29] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*. 726–735.

[30] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. 2019. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *WWW*. 2091–2102.

[31] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: a survey. *Comput. Surveys* 55, 5 (2022), 1–37.

[32] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. 2022. Hypergraph contrastive collaborative filtering. In *SIGIR*. 70–79.

[33] Lianghao Xia, Chao Huang, and Chuxu Zhang. 2022. Self-supervised hypergraph transformer for recommender systems. In *KDD*. 2100–2109.

[34] Lianghao Xia, Yong Xu, Chao Huang, Peng Dai, and Liefeng Bo. 2021. Graph meta network for multi-behavior recommendation. In *SIGIR*. 757–766.

[35] Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. 2020. Heterogeneous network representation learning: A unified framework with survey and benchmark. *Transactions on Knowledge and Data Engineering* (2020).

[36] Liang Yang, Fan Wu, Zichen Zheng, Bingxin Niu, Junhua Gu, Chuan Wang, Xiaochun Cao, and Yuanfang Guo. 2021. Heterogeneous Graph Information Bottleneck.. In *IJCAI*. 1638–1645.

[37] Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge Graph Contrastive Learning for Recommendation. In *SIGIR*.

[38] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *NeurIPS* 33 (2020), 5812–5823.

[39] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021. Self-Supervised Multi-Channel Hypergraph Convolutional Network for Social Recommendation. In *WWW*. 413–424.

[40] Shengyu Zhang, Lingxiao Yang, Dong Yao, Yujie Lu, Fuli Feng, Zhou Zhao, Tat-seng Chua, and Fei Wu. 2022. Re4: Learning to Re-contrast, Re-attend, Re-construct for Multi-interest Recommendation. In *WWW*. 2216–2226.

[41] Weifeng Zhang, Jingwen Mao, Yi Cao, and Congfu Xu. 2020. Multiplex graph neural networks for multi-behavior recommendation. In *CIKM*. 2313–2316.

[42] Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. 2021. Personalized Transfer of User Preferences for Cross-domain Recommendation. *arXiv preprint arXiv:2110.11154* (2021).

[43] Ding Zou, Wei Wei, Ziyang Wang, Xian-Ling Mao, Feida Zhu, Rui Fang, and Dangyang Chen. 2022. Improving knowledge-aware recommendation with multi-level interactive contrastive learning. In *CIKM*. 2817–2826.