Applied Soft Computing Journal 173 (2025) 112934

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Heterogeneous collaborative filtering contrastive learning for social recommendation

Chaojun Meng [a], Changfan Pan [a], Hongji Shu [a], Qing Wang [b], Hanghui Guo [a], Jia Zhu [c,*]

[a] School of Computer Science and Technology, Zhejiang Normal University, Jinhua, China
[b] Department of Health Outcomes and Biomedical Informatics, College of Medicine, University of Florida, FL, USA
[c] Zhejiang Key Laboratory of Intelligent Education Technology and Application, Zhejiang Normal University, Jinhua, China

ARTICLE INFO

Keywords:
Recommendation system
Contrastive learning
Graph Neural Network

ABSTRACT

Collaborative filtering methods based on Graph Neural Networks (GNNs) have gained increasing popularity in recommendation systems. These methods enhance the representation of users and items by leveraging the information of graph structure from interaction data, improving recommendation performance. However, they often face limitations due to the data sparsity issue that is common in recommendation systems. In the constructed user–item heterogeneous bipartite graph, sparse interaction data leads to a scarcity of neighbor nodes impeding the acquisition of sufficient collaborative signals via the message-passing mechanism among these neighbor nodes. We have observed that users and items can be grouped according to characteristic similarities. These groups' common feature information can serve as supplementary data to aid in the embedding learning. So, we present the Heterogeneous Collaborative Filtering Contrastive Learning (HCFCL) method, which aims to extract two types of heterogeneous collaborative signals from interaction data: those based on neighbor nodes and those based on group features. Specifically, we design an embedding generative hypergraph network to extract group common feature information founded on the heterogeneous bipartite graph. The group common feature information is transferred via a meta network and personalized bridge functions according to individual characteristics. Additionally, the HCFCL model, combined with contrastive learning, captures the consistency of the heterogeneous collaborative signals to enhance representation. The experiment demonstrates the superior performance of the HCFCL model compared to other methods evaluated on three public datasets, demonstrating excellent and stable performance in mitigating the data sparsity issue.

1. Introduction

Recommendation systems help users sift through large volumes of data, efficiently guiding them to content that aligns with their preferences by filtering out irrelevant information. Consequently, they have found widespread application across various domains, including e-commerce platforms [1,2], social media platforms [3], and video streaming services [4]. Currently, a substantial amount of research is dedicated to effectively enhancing the performance of recommendation systems. The majority of this researches revolve around collaborative filtering. Collaborative filtering is a method employed to uncover user preferences through the mining of interaction data, and to subsequently recommend items that align with these preferences.

Traditional collaborative filtering methods [5,6] use pre-set user and item embedding matrices to construct a new interaction matrix through carefully designed mechanisms. Then, the new interaction matrix approximates the real user–item interaction matrix to aid in learning collaborative signals. However, these methods do not fully encode the collaborative signals to the embeddings of users and items, thus failing to achieve optimal collaborative effects [7]. In recent years, the ascendancy of Graph Neural Networks (GNNs) has catalyzed the development of various GNNs-based collaborative filtering methods in recommendation systems due to their exceptional performance [8–10]. These methods propagate information across the user–item heterogeneous bipartite graph [11], imbuing node embeddings with collaborative signals effectively.

Although GNN-based collaborative filtering methods are highly effective, they are limited by data sparsity issue and imbalanced distribution. These problems result in some nodes having too few neighbors, leading to insufficient information being available. This limitation hampers the representation learning of these nodes, preventing them from

* Corresponding author.
E-mail addresses: mengchaojun@zjnu.edu.cn (C. Meng), changfanpan@zjnu.edu.cn (C. Pan), shj451148969@zjnu.edu.cn (H. Shu), wang.qing@ufl.edu (Q. Wang), ghh1125@zjnu.edu.cn (H. Guo), jiazhu@zjnu.edu.cn (J. Zhu).

https://doi.org/10.1016/j.asoc.2025.112934
Received 30 June 2024; Received in revised form 27 January 2025; Accepted 21 February 2025
Available online 1 March 2025

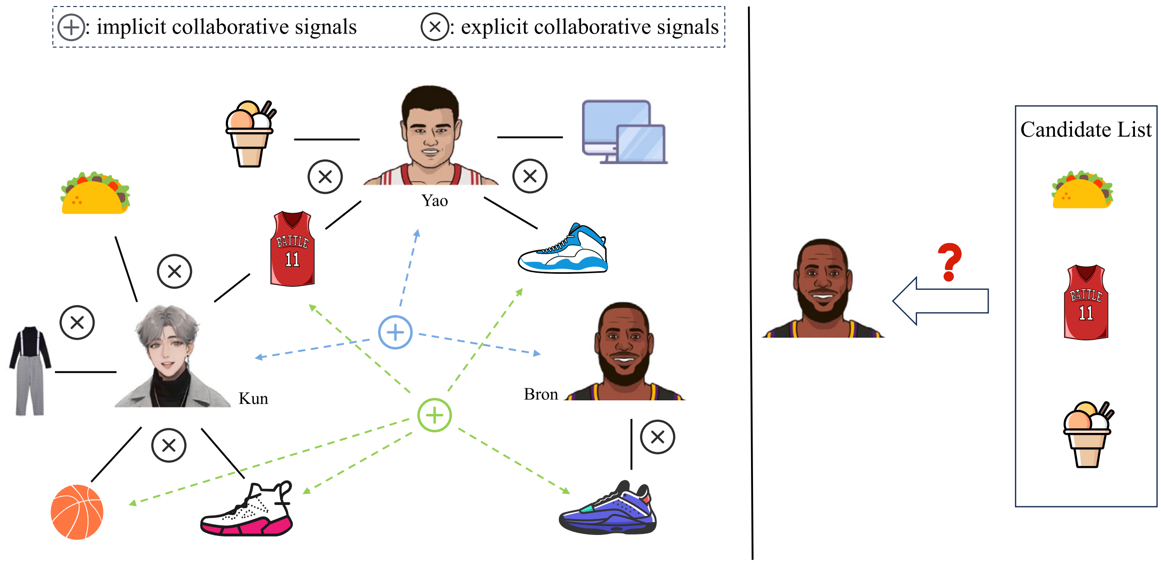


Fig. 1. Relying only on explicit collaborative signals from a small number of direct interactions, it is difficult to mine Bron's preference for basketball-related items, and thus it is impossible to recommend basketball uniforms to Bron from the candidate list. Mining implicit collaborative signals can solve this problem. However, the group that is also interested in basketball sport in the bipartite graph is not connected to Bron, and traditional GNN-based recommendation methods cannot mine implicit collaborative signals.

achieving the desired effect. Iterating through multiple graph propagation layers can expand the receptive field of message passing and aggregation mechanisms to mitigate the effects of data sparsity issue. Nevertheless, the inherent smoothing issue of GNNs restricts the number of propagation layer iterations, limiting the effectiveness of this approach in addressing data sparsity issue [12]. Thus, this method provides only a partial solution to the problem.

To alleviate the data sparsity issue, it is essential to deeply mine collaborative signals within interaction data. One key aspect of this data involves direct interactions between users and items, which serve as an intuitive and genuine mapping of user preferences to item characteristics. We refer to the user preferences and item characteristics contained in these direct interactions as explicit collaborative signals. Explicit collaborative signals are distributed among the neighboring nodes directly connected to each node, forming a local subgraph in the heterogeneous bipartite graph. Thus, capturing these explicit collaborative signals necessitates focusing on the local information within the heterogeneous bipartite graph.

On the other hand, there may be certain associations between nodes of the same type, even if there are no direct interactions between them. For example in Fig. 1, users Kun and Bron have both purchased basketball-related items. Following the principle that similar items and users tend to cluster together, Kun and Bron can be grouped based on their common interest. By mining the common features within such a group, their embeddings can be enrich with additional information. We refer to the common feature exhibited by the group as latent collaborative signals. Mining latent collaborative signals helps alleviate the problem of insufficient explicit collaborative signals caused by sparse neighboring nodes.

However, fully mining latent collaborative signals is challenging. Nodes within the same group may not be directly connected, and even if a path exists between them, it might be too long to effectively mine latent collaborative signals through the message passing mechanism. Additionally, group features in latent collaborative signals are generalizations of individual features, while individual features are specializations of group features. There are differences between individual features and group features. For instance in Fig. 1, users Kun and Bron in the same group might both purchase basketball shoes, but due to individual differences, the specific styles of basketball shoes they purchase may vary. Therefore, it is necessary to transfer group common feature information for individuals. Essentially, the primary issues that need to be resolved are: (1) How to effectively acquire latent collaborative signals; (2) How to fulfill personalized transfer of the group common feature information contained in the latent collaborative signals to individuals.

To tackle these challenges, we propose the Heterogeneous Collaborative Filtering Contrastive Learning (HCFCL) model. The HCFCL model initially captures explicit collaborative signals within the heterogeneous bipartite graph. Building on this, it employs an embedding generative hypergraph network to construct user homogeneous graphs and item homogeneous graphs for mining latent collaborative signals. Then, a feature knowledge personalization transfer network is designed to transfer the group common feature information contained in the latent collaborative signals. This network uses a meta network to extract feature knowledge and employs a personalized bridge function to map feature knowledge to individual features, thereby achieving the customization of group common feature information. Finally, contrastive learning is used to reduce the disparity between the two types of heterogeneous collaborative signals, integrating both types of signals into the user and item embeddings.

The contributions of this study can be outlined as follows:

- We innovatively propose a multi-view representation learning framework that combines different graph networks based on the characteristics and relation of heterogeneous collaborative signals to learn user and item representations.
- We propose the HCFCL model innovatively employ a meta network to assist the personalized bridge function in transferring the group common feature information. Through contrastive learning, it effectively integrates heterogeneous collaborative signals from different views to enhance embedding learning.
- We validate the effectiveness of the HCFCL model using three publicly available recommendation system datasets. Our results clearly demonstrate that the HCFCL model achieves state-of-the-art performance, significantly outperforming other baseline methods.

2. Related work

2.1. Graph-based recommender systems

Graph Neural Networks (GNNs) excel in learning graph data embeddings due to their message passing and aggregation mechanisms, which effectively capture graph structure information. Many graph-based recommender systems leverage the characteristics of GNNs to capture collaborative signals, thereby optimizing the representation of users and items. For instance, LightGCN [13] and KGAT [14] construct bipartite graph, refining user and item node embeddings through multi-layer message propagation.

To further obtain richer information, several works such as NGCF [7], PinSage [15], MB-GMN [16], and GraphDA [17] have proposed message passing and aggregation mechanisms based on multi-hop interaction topologies to capture collaborative filtering signals with high-order connectivity. Inspired by the properties of hypergraphs, some works solve the problem that graph convolutional networks are difficult to model high-order dependencies by constructing hypergraph structures. For example, MHCN [18] divides complex social relationships, and each social relationship is modeled through different hypergraph convolutions. DHCF [19] uses hypergraphs to model high-order correlations between complex entity relationships between users and between items.

Other works focus on mining fine-grained information in user–item interaction data [20,21]. Among these approaches, recommendation methods that leverage heterogeneous graphs have garnered significant attention for their outstanding performance. Heterogeneous graph-based recommendation methods adopt different processing methods according to different types of nodes or edges, aiming to encode different types of information and thus mine the rich semantics in heterogeneous relationships. For instance, MEIRec [22] proposed a meta-path guided information aggregation mechanism that can better learn graph structure information while enriching node neighbor information. The model designs different aggregation functions based on the characteristics of different types of neighbor information to capture different types of semantic information in a fine-grained manner. GHCF [23] models heterogeneous behavioral relationships of multiple types of interactions, taking each behavior as a subtask, and finally optimizes the model through multi-task learning.

Essentially, graph-based recommender systems mainly acquire information about relevant nodes in the graph in various ways to deeply mine the collaborative signals hidden in interaction data, and finally aggregate the acquired information to optimize the embedding representation of target nodes.

2.2. Meta learning

The aim of meta learning is to identify commonalities across different tasks for a model, allowing it to perform well on various similar tasks. Existing meta learning methods fall into three main categories: (1) Black-box amortized methods [24,25]: These methods utilize a black-box meta learner to acquire the model's parameters. (2) Gradient-based methods [26,27]: They learn an optimal initialization parameters of model that can efficiently adapt to new tasks with minimal updates for gradient. (3) Non-parametric methods [28,29]: These methods employ suitable distance metrics to evaluate samples obtained from both parametric meta learners and non-parametric learners. Their goal is to maximizing the separation between dissimilar samples while minimize the distance between similar samples. Some works have applied meta learning to improve the effectiveness of recommendation systems. These work primarily focus on the cold-start problem [30,31] and cross-domain recommendation [32]. Our research focuses on using meta learning to address the differences in heterogeneous collaborative signals.

2.3. Contrastive learning in recommendation systems

In recommender systems, contrastive learning is widely employed to enhance data by aligning feature representations from different views, making it an effective self-supervised learning method [33,34]. Numerous works have leveraged contrastive learning to alleviate data sparsity issues, achieving notable results [35,36]. SGL [37] utilizes techniques such as random walks, edge dropout, or node dropout to generate diverse bipartite graphs for data augmentation. KGCL [38] not only applies dropout operations in the knowledge graph, but also introduces a knowledge-aware contrastive approach to compare and enhance the node representations learned in the interaction graph and the knowledge graph. These methods fall under the local–local level graph structure contrastive approaches.

To integrate global information of graph structures into node representations, some methods have proposed local–global level graph structure contrastive approaches. HGCL [39] constructs homogeneous graphs for user and item node types. It computes the average mutual information between local node representations and graph-level summary representations within each homogeneous graph. HCCF [40] designs a hypergraph structure to capture global collaborative relations, enhancing self-supervision through local–global contrastive learning. In our research, we capture the consistency of heterogeneous collaborative filtering signals across multiple views through contrastive learning, thereby achieving the fusion of heterogeneous information.

3. Preliminary

In a recommendation scenario, the graph $G_{He} = \{V, A\}$ represents a heterogeneous bipartite graph constructed from the interaction data. Here, V represents the node set in a heterogeneous bipartite graph. In the node set V, it is divided into two subsets according to the node type: user set V_u and item set V_i. $E_{He^u} \in \mathbb{R}^{q\times d}$ denotes the embedding matrix of the user set V_u, and $E_{He^i} \in \mathbb{R}^{s\times d}$ is the embedding matrix of the item set V_i. Here, q and s denote the amount of users and items, respectively, while d is the embedding dimensionality. A is the interaction matrix that indicates the presence of interaction relationships between user and item nodes. $A_{i,j}$ is set to 1 when user u_i interacts with item v_j; otherwise, $A_{i,j} = 0$.

The graph $G_{Hy} = \{X, H\}$ represents a homogeneous hypergraph, where $x \in X$ is a collection of nodes of the same type. In this work, two types of homogeneous graphs are derived from graph G_{He}: user homogeneous hypergraph G_{Hy^u} and item homogeneous hypergraph G_{Hy^i}. $H \in \mathbb{R}^{k\times g}$ denotes the affiliation relationships between node set X and the groups set Y. If node x_i belong to groups y_j, the entry $H_{i,j}$ is set as 1 and $H_{i,j} = 0$ otherwise.

In this work, the heterogeneous bipartite graph G_{He} is used as input. Founded on the heterogeneous bipartite graph G_{He}, a user homogeneous hypergraph G_{Hy^u} and an item homogeneous hypergraph G_{Hy^i} are constructed. Implicit collaborative signals are mined on the homogeneous hypergraph. Finally, the two heterogeneous collaborative signals are fused through contrastive learning for data augmentation. The prediction function then outputs the likelihood of interaction between the user and the item based on their respective representation.

4. Methodology

As illustrated in Fig. 2, the framework is primarily divided into three parts: (1) **Heterogeneous Collaborative Signal Mining**: It mainly comprises a graph convolutional encoder and an embedding generative hypergraph network. These components are designed to mine heterogeneous collaborative signals on both heterogeneous bipartite graphs and homogeneous hypergraphs, facilitating representation learning. (2) **Feature Knowledge Personalized Transfer Network**: This part is composed of a meta network and a personalized bridge function to optimize the representation. The meta network can dynamically generate the parameters of the personalized bridge function based on the representation to assist the personalized bridge function in transferring the group common feature information. (3) **Heterogeneous Collaborative Signal Contrastive Learning**: It achieves the fusion of heterogeneous collaborative signals by capturing the feature consistency in multiple views.

The following sections will provide a detailed introduction to the various components of the HCFCL framework.

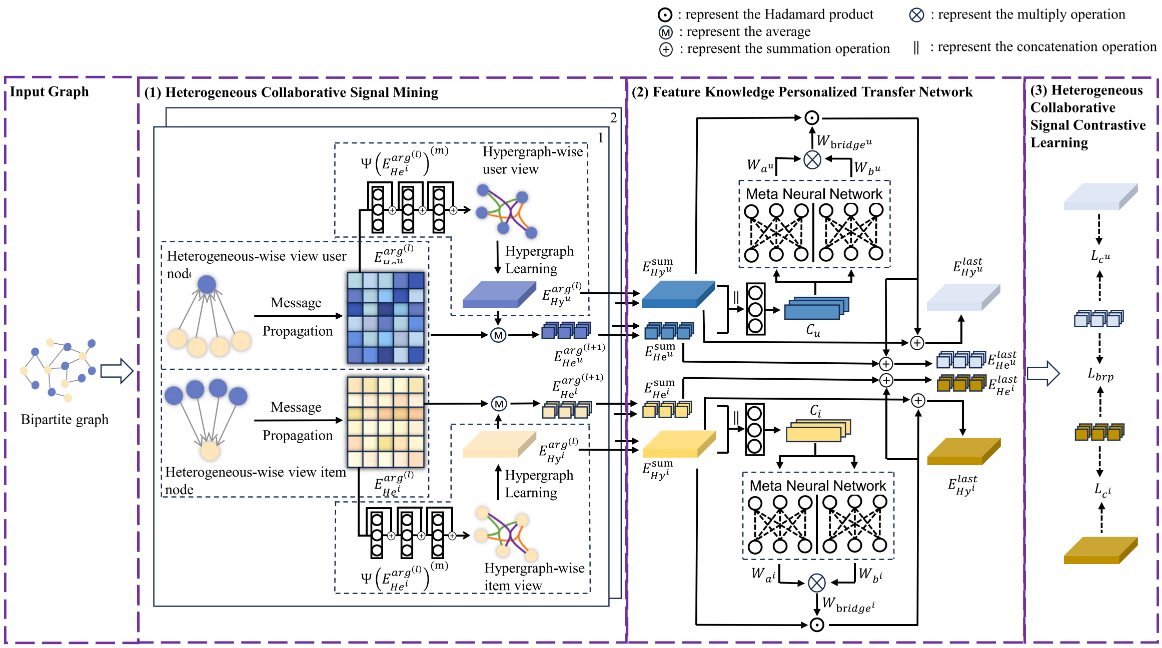


Fig. 2. Illustration of the HCFCL model framework. It contains three parts: (1) The heterogeneous-wise view (Section 4.1.1) and hypergraph-wise view (Section 4.2.1) are used to mine heterogeneous collaborative signals; (2) Involving feature knowledge extraction (Section 4.2.1) and feature knowledge personalized transfer (Section 4.2.2) to achieve representation optimization; (3) Heterogeneous Collaborative Signal Contrastive Learning for data augmentation.

4.1. Heterogeneous collaborative signals mining

There are abundant implicit collaborative signals in the interaction data, which can serve as additional information to make up for the lack of explicit collaborative signals caused by sparsity. There is a certain connection between these two heterogeneous collaborative signals because the user and item characteristics expressed in the explicit collaborative signal determine the user and item's affiliation with the group expressed in the implicit collaborative signal. However, due to their heterogeneity, different operations need to be performed according to their respective characteristics to complete the mining of heterogeneous collaborative signals. Therefore, in this section, we first introduce how to capture explicit collaborative signals, then describe the mining of implicit collaborative signals, and finally introduce how to combine the heterogeneous collaborative signals to complete the representation learning.

4.1.1. Capturing explicit collaborative signals

The graph convolution operations demonstrate a strong ability to capture explicit collaborative signals [7,15]. Complex operations will introduce more parameters, which will lead to overfitting when data is scarce. Therefore, in our work, we employ the basic GCN as the graph encoder to learn richer representations of nodes. Inspired by the LightGCN [13], the graph encoder removes the redundant operations of feature transformation to further simplify the model. Thus, the node update in the explicit collaborative signal capture part is as follows:

$$E_{He}^{arg^{(l)}} = \sigma\left(\hat{A}_{He} E_{He}^{(l)}\right), \hat{A}_{He} = D_{He}^{-\frac{1}{2}} A_{He} D_{He}^{-\frac{1}{2}} \tag{1}$$

$E_{He}^{agg^{(l)}} \in \mathbb{R}^{(q+s)\times d}$ represents the heterogeneous-wise node information aggregation embedding matrix obtained by aggregating neighbor node information from the lth layer node embedding matrix $E_{He}^{(l)} \in \mathbb{R}^{(q+s)\times d}$. Specifically, $E_{He}^{(0)}$ is initialized randomly from a normal distribution. $\hat{A}_{He} \in \mathbb{R}^{(q+s)\times(q+s)}$ represents the normalized adjacency matrix computed from the interaction matrix $A_{He} \in \mathbb{R}^{(q+s)\times(q+s)}$. $D_{He} \in \mathbb{R}^{(q+s)\times(q+s)}$ denotes the diagonal degree matrix of the heterogeneous graph. $\sigma(\cdot)$ denotes the PReLU activation function. The self-information incorporation of the nodes is not considered during the message aggregation to avoid additional interference from redundant information.

4.1.2. Mining implicit collaborative signals based on hypergraph learning

Data sparsity can result in some user or item nodes having too few neighbor nodes, which means that explicit collaborative signals cannot provide sufficient information for learning node representations. Therefore, it is necessary to mine implicit collaborative signals as complementary information to enrich the representations. Mining implicit collaborative signals require partitioning nodes that belong to the same group. In a hypergraph, a hyperedge can connect an unlimited number of nodes. In this scenario, each hyperedge connects nodes belonging to the same group, forming a hypergraph structure. The group common feature information is then aggregated into the node representation through the hypergraph network. However, the construction of hypergraph structures usually relies on preset primitive connections, thus ignoring the connection between individual characteristics and group features. Therefore, we propose an embedding generative hypergraph network based on the basic hypergraph network to construct the hypergraph structure and implement hypergraph representation learning.

The general representation learning paradigm of a hypergraph network is as follows:

$$X^{(l+1)} = \sigma\left(HWH^T X^{(l)} P\right) \tag{2}$$

$X^{(l+1)} \in \mathbb{R}^{k\times d}$ denotes the node embedding matrix of the $(l+1)$th layer hypergraph, derived from the lth layer node embedding matrix $X^{(l)} \in \mathbb{R}^{k\times d}$ through hypergraph convolution operations. $H \in \mathbb{R}^{k\times g}$ represents the incidence matrix of the hyperedges. $P \in \mathbb{R}^{d\times d}$ is a learnable parameter matrix. $W \in \mathbb{R}^{g\times g}$ represents the diagonal matrix of hyperedge weights. Since W is a diagonal matrix, $W = W^{\frac{1}{2}}W^{\frac{1}{2}}$, and $W^{\frac{1}{2}}$ is also a diagonal matrix. Additionally, because $W^T = W$, we have $W^{\frac{1}{2}} = (W^{\frac{1}{2}})^T$, so $W = W^{\frac{1}{2}}(W^{\frac{1}{2}})^T$. Thus, the node update paradigm for the hypergraph neural network is as follows:

$$X^{(l+1)} = \sigma\left(\left(HW^{\frac{1}{2}}\right)\left(HW^{\frac{1}{2}}\right)^T X^{(l)} P\right) \tag{3}$$

Hyperedges connect individual nodes with similar characteristics to form groups. The characteristics information has been injected into the node representation by mining explicit collaborative signals. Therefore, the node representation matrix has a certain intrinsic connection with

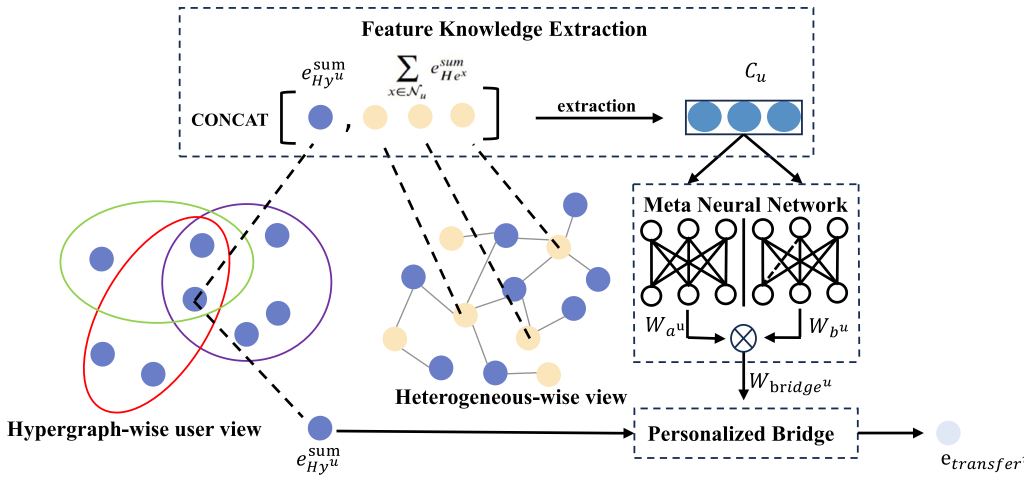


Fig. 3. Work flow of the feature knowledge personalized transfer network. Different colored circles in the hypergraph-wise user view represent different hyperedges.

the hyperedge incidence matrix. This inherent connection can be represented by constructing a mapping relationship $\Psi(\cdot)$ from the node representation matrix X to the hyperedge incidence matrix H and the hyperedge weight diagonal matrix W:

$$\Psi(X) \rightarrow HW^{\frac{1}{2}}, X^{(l+1)} = \sigma\left(\left(\Psi\left(X^{(l)}\right)\right)\left(\Psi\left(X^{(l)}\right)\right)^{T} X^{(l)} P\right) \tag{4}$$

Drawing inspiration from the gating mechanism used in GRUs [41], the gating mechanism can act as a switch to control the affiliation relationship from node to group. So the hypergraph structure construction network $\Psi(X)$ adopts the gating mechanism to fit the mapping relationship. Additionally, the network's ability to mine implicit collaborative signals is enhanced by stacking layers of the hypergraph structure construction network through a residual connection operation. Formally, the hypergraph structure construction network $\Psi(X)$ is:

$$\begin{aligned} \Psi(X)^{(0)} &= \sigma\left(XW_g + b_g\right), \\ \Psi(X)^{(l+1)} &= \sigma\left(\Psi(X)^{(l)} W_c\right) + \Psi(X)^{(l)} \rightarrow HW^{\frac{1}{2}} \end{aligned} \tag{5}$$

where $W_g \in \mathbb{R}^{d\times g}$ is learnable gating matrix. $b_g \in \mathbb{R}^{k\times 1}$ is bias. $W_c \in \mathbb{R}^{g\times g}$ represents the learnable parametric matrix. The activation function $\sigma(\cdot)$ adopts LeakyReLU to handle non-linear transformation.

Therefore, user and item nodes conduct representation learning through the embedding generative hypergraph network to incorporate implicit collaborative signals information. The formal representation of learning is:

$$\begin{aligned} E^{arg^{(l)}}_{Hy^u} &= \sigma\left(\left(\Psi\left(E^{arg^{(l)}}_{He^u}\right)^{(m)}\right)\left(\Psi\left(E^{arg^{(l)}}_{He^u}\right)^{(m)}\right)^{T} E^{arg^{(l)}}_{He^u} P\right), \\ E^{arg^{(l)}}_{Hy^i} &= \sigma\left(\left(\Psi\left(E^{arg^{(l)}}_{He^i}\right)^{(m)}\right)\left(\Psi\left(E^{arg^{(l)}}_{He^i}\right)^{(m)}\right)^{T} E^{arg^{(l)}}_{He^i} P\right) \end{aligned} \tag{6}$$

$E^{arg^{(l)}}_{He^u} \in \mathbb{R}^{q\times d}$ and $E^{arg^{(l)}}_{He^i} \in \mathbb{R}^{s\times d}$ are the lth layer heterogeneous-wise node information aggregation embedding matrices from Section 4.1.1, serving as the input to the lth layer of the embedding generative hypergraph network. $E^{arg^{(l)}}_{Hy^u} \in \mathbb{R}^{q\times d}$ and $E^{arg^{(l)}}_{Hy^i} \in \mathbb{R}^{s\times d}$ are the hypergraph-wise user node information aggregation embedding matrix and the hypergraph-wise item node information aggregation embedding matrix, respectively. $m \in \mathbb{R}$ represents the layers of the hypergraph structure construction network and is equal to 3 in our work. $\sigma(\cdot)$ uses LeakyReLU as the activation function.

4.1.3. Heterogeneous collaborative signals aggregation

In the heterogeneous graph, each node iteration can aggregate different semantic information from heterogeneous relationships. Through multiple iterations, the model's receptive field is expanded, allowing higher-level embeddings to gather heterogeneous information from multi-hop connections. In the HCFCL model, there are two types of heterogeneous semantic information: explicit collaborative signals in the heterogeneous-wise view and implicit collaborative signals in the hypergraph-wise view. The iterative fusion method of these two types of heterogeneous semantic information is:

$$E^{(l+1)}_{He} = \alpha E^{arg^{(l)}}_{He} + \beta E^{arg^{(l)}}_{Hy} \tag{7}$$

here, $E^{(l+1)}_{He} \in \mathbb{R}^{(q+s)\times d}$ integrates the two types of semantic information and serves as the heterogeneous graph node embedding matrix for the $(l+1)$th layer. This matrix is computed by a proportional summation of two embedding matrices from (l)th layer: the heterogeneous-wise node information aggregation embedding matrix $E^{arg^{(l)}}_{He} \in \mathbb{R}^{(q+s)\times d}$ and the hypergraph-wise node information aggregation embedding matrix $E^{arg^{(l)}}_{Hy} \in \mathbb{R}^{(q+s)\times d}$. α and β are the proportionality coefficients. This fusion method reduces computational complexity while retaining semantic information from both perspectives. By analyzing the same data from different angles, it creates a richer and more comprehensive representation, effectively addressing data sparsity. The two perspectives are complementary rather than conflicting, with their proportional contributions adjustable based on specific application scenarios. For example, when user interactions are sparse, more weight can be given to implicit collaborative signals. In this work, equal importance is assigned to both perspectives, setting α and β to $\frac{1}{2}$.

To further capture information from each layer, from lower to higher orders, in different views, the overall node embedding matrices for each view are generated as follows:

$$E^{sum}_{He} = \sum_{l=0}^{L} \frac{E^{arg^{(l)}}_{He}}{\left\| E^{arg^{(l)}}_{He} \right\|}, E^{sum}_{Hy} = \sum_{l=0}^{L} \frac{E^{arg^{(l)}}_{Hy}}{\left\| E^{arg^{(l)}}_{Hy} \right\|} \tag{8}$$

4.2. Feature knowledge personalized transfer network

After mining heterogeneous collaborative filtering signals, excellent representations can be learned. However, there are differences between the group common feature information and the individual characteristics in the implicit collaborative signal. As shown in Fig. 1, different individuals who like basketball shoes may have different preferences due to variations in shoe styles. This difference can lead to biases in representation learning. Therefore, the personalized transfer of group common feature information is essential to optimize the representation. The detailed workflow of this chapter is shown in Fig. 3.

4.2.1. Feature knowledge extraction

To achieve the personalized transfer of group common feature information, it is need to filter out irrelevant information from the implicit collaborative signals and extract feature knowledge related to individuals. Additionally, the user preferences and item characteristics expressed in the explicit collaborative signals are strongly related to individuals. Extracting feature knowledge from this part of the information can increase the association between feature knowledge and individual characteristics, thereby reducing the difficulty of personalizing feature knowledge transfer. Specifically, the extraction of feature knowledge is as follows:

$$\begin{aligned} C_u &= \mathrm{CONCAT}\left(E^{sum}_{Hy^u}, \sum_{x\in\mathcal{N}_u} e^{sum}_{He^x}\right) W_u + b_u, \\ C_i &= \mathrm{CONCAT}\left(E^{sum}_{Hy^i}, \sum_{x\in\mathcal{N}_i} e^{sum}_{He^x}\right) W_i + b_i, \end{aligned} \tag{9}$$

$C_u \in \mathbb{R}^{q\times d}$ and $C_i \in \mathbb{R}^{s\times d}$ denote the feature knowledge of users and items, respectively. Feature knowledge is composed of two concatenated parts: the hypergraph-wise overall node embeddings representing the part of implicit collaborative signals, and the neighboring node embeddings of the heterogeneous-wise overall nodes representing the part of the explicit collaborative signals. $W_i \in \mathbb{R}^{2d\times d}$ and $W_u \in \mathbb{R}^{2d\times d}$ are trainable parameter matrices for items and users, respectively, which extract knowledge from the heterogeneous collaborative signals in two different views and map them into the same feature knowledge space. $b_u \in \mathbb{R}^{d\times 1}$ and $b_i \in \mathbb{R}^{d\times 1}$ are biases. $\mathrm{CONCAT}(\cdot,\cdot)$ denotes the concatenation operation.

4.2.2. Feature knowledge personalized transfer

To achieve the personalized transfer of feature knowledge, we designed a function to bridge the feature knowledge from the group domain to the individual domain. Intuitively, the individual domain for different users/items varies, necessitating different bridge functions for different users/items to enable personalized transfer. Inspired by the personalized bridge function [42], the HCFCL model employs a meta network to generate customized transformation matrices based on the extracted feature knowledge, facilitating adaptive feature knowledge transfer. These generative matrices act as the parameter matrices for the personalized bridge function. Take the user nodes for example, the proposed meta network is:

$$W_{bridge^u} = \mathrm{MN}\left(C_u\right) \to W_{a^u} W_{b^u} \tag{10}$$

where $W_{bridge} \in \mathbb{R}^{m\times d}$ is the parameter matrix of the personalized bridge function, which is obtained by processing the feature knowledge of users through a meta network $\mathrm{MN}(\cdot)$. Users and items have their respective meta networks for personalization. To reduce the parameter numbers in training, the high-dimensional complex feature knowledge matrix C_u in the meta network $\mathrm{MN}(\cdot)$ is decomposed by low-rank decomposition to extract the main feature information from the knowledge matrix, removing noise and redundant information. Finally, W_{a^u} and W_{b^u} are multiplied to generate the parameter matrix of the personalized bridge function. The specific process of meta network $\mathrm{MN}(\cdot)$ is shown below:

$$W_{a^u} = f^c_{meta}\left(C_a\right),\ C_a = C_u O_a + b W_{b^u} = f^c_{meta}\left(C_b\right),\ C_b = O_b C_u + b \tag{11}$$

$$f^c_{meta}(C) = \begin{cases} \mathrm{MLP_a}(C) & if\ \ C \in \mathbb{R}^{q\times t} \\ \mathrm{MLP_b}(C) & if\ \ C \in \mathbb{R}^{t\times d} \end{cases} \tag{12}$$

where $O_a \in \mathbb{R}^{d\times t}$ and $O_b \in \mathbb{R}^{t\times d}$ denote the decomposition matrices of the feature knowledge dimensions, which decompose the feature knowledge matrix into two matrices $C_a \in \mathbb{R}^{q\times t}$ and $C_b \in \mathbb{R}^{t\times d}$, respectively. The meta function $f^c_{meta}(\cdot)$ consists of two c-layer fully connected neural network $\mathrm{MLP_a}(\cdot)$ and $\mathrm{MLP_b}(\cdot)$. It takes these decomposed matrices as input and outputs matrices $W_a \in \mathbb{R}^{q\times t}$ and $W_b \in \mathbb{R}^{t\times d}$. The maximum feature information of the decomposed matrices W_a and W_b is limited by the rank t of the matrix; the larger the t, the greater the maximum feature information, but also the more noise and redundant information. By this way, the complexity of meta network $\mathrm{MN}(\cdot)$ changes from $\mathcal{O}(cqd^2)$ to $\mathcal{O}((c-1)(q+d)t^2 + 2qdk)$. Since $t \ll d$, the complexity can be further expressed as $\mathcal{O}((ck-t+2q)kd)$. The W_{bridge} obtained by multiplying W_a and W_b is used as the parameter matrix to construct the personalized bridge function as follows:

$$E_{transfer^u} = \sigma\left(W_{bridge^u} \odot E^{sum}_{Hy^u}\right) \tag{13}$$

$$E^{last}_{Hy^u} = E^{sum}_{Hy^u} + E_{transfer^u}, E^{last}_{He^u} = E^{sum}_{He^u} + E_{transfer^u} \tag{14}$$

where $E_{transfer^u} \in \mathbb{R}^{q\times d}$ represents the customized embedding matrix of common information after personalized transfer by the hypergraph-wise user node embedding matrix $E^{sum}_{Hy^u}$. $\odot$ denotes the Hadamard product operation. $\sigma(\cdot)$ adopts the PReLU activation function. Then, the customized embedding matrix $E_{transfer^u}$ is fused with the node embedding matrix $E^{sum}_{Hy^u}$ and $E^{sum}_{He^u}$, resulting in the final embedding matrix for the hypergraph-wise user node $E^{last}_{Hy^u} \in \mathbb{R}^{q\times d}$ and the heterogeneous-wise user node $E^{last}_{He^u} \in \mathbb{R}^{q\times d}$. The processing for the item nodes is similar to the above process.

4.3. Heterogeneous collaborative signal contrastive learning

In the above process, the information carried by the collaborative signals in different views differs, leading to discrepancies in the embeddings of the same nodes in hypergraph-wise and heterogeneous-wise views. The HCFCL model introduces contrastive learning, which aligns the embeddings of nodes from hypergraph-wise and heterogeneous-wise views, thereby reducing the discrepancies. The aligned embeddings of nodes can integrate the heterogeneous collaborative signals from both views, alleviating the problem of data sparsity and enhancing data augmentation.

We consider the embeddings of the same user/item node in hypergraph-wise and heterogeneous-wise views as positive pairs $\left(e^{last}_{Hy^i}, e^{last}_{He^i}\right)$, while the embeddings of different user/item nodes are treated as negative pairs $\left(e^{last}_{Hy^i}, e^{last}_{He^j}\right)$. i and j denote the index of node $(i \neq j)$. In contrastive learning, the embeddings of nodes in positive pairs gradually become closer in the space, thus reducing the discrepancies between embeddings of nodes from different views. Conversely, the embeddings of nodes in negative pairs move further apart, ensuring each node has its unique semantic features. The HCFCL model employs an InfoNCE-based contrastive learning method to calculate the loss, as follows:

$$\mathcal{L}_{c^u} = \sum_{z\in U} -\log \frac{\exp\left(\frac{\cos\left(e^{last}_{Hy^z}, e^{last}_{He^z}\right)}{\tau}\right)}{\sum_{j\in U}\exp\left(\frac{\cos\left(e^{last}_{Hy^z}, e^{last}_{He^j}\right)}{\tau}\right)} \tag{15}$$

where $e^{last}_{Hy^i}$ and $e^{last}_{He^i}$ represent the embeddings of the user node with index i in embedding matrices $E^{last}_{Hy^u}$ and $E^{last}_{He^u}$, respectively. The temperature coefficient τ regulates the model's sensitivity of negative samples. The loss of contrastive learning for the corresponding item nodes is calculated in the same manner as described above.

4.4. Model prediction and optimization

In the HCFCL model, the likelihood of interaction preference between a user and an item is assessed using the inner product of their respective vectors, as shown below:

$$p(u,i) = e^{last}_{He^u} \cdot e^{last}_{He^i} \tag{16}$$

$p \in \mathbb{R}$ is the interaction preference probability, with a higher value indicating a greater probability of interaction between the user and

Table 1
Analysis of the experimental datasets.

Dataset	User	Item	Interaction	Density
Yelp	161 305	114 852	957 923	5.17e^{-5}
Epinions	15 210	233 929	630 391	1.77e^{-4}
Ciao	6776	101 415	265 308	3.86e^{-4}

the item. The top k items with the highest probabilities are selected as the user's recommended results to achieve information filtering effect. $e^{last}_{He^u} \in \mathbb{R}^{d\times 1}$ and $e^{last}_{He^i} \in \mathbb{R}^{d\times 1}$ represent the vector for the corresponding user and item in the node information aggregation embedding matrix with the heterogeneous-wise view, respectively. And we adopt the Bayesian Personalized Ranking (BPR) loss function to optimize model:

$$\mathcal{L}_{brp} = \sum_{(u,i^+,i^-)\in D} -\ln\left(\text{sigmoid}\left(p\left(u,i^+\right) - p\left(u,i^-\right)\right)\right) \tag{17}$$

here, u represents the user node in the triplet set D, i^+ is the item node that have already established an interaction with the user node u. On the other hand, i^- represents the item nodes obtained through random sampling, which have not shown any interaction with the user node u in the records. The overall loss function is derived by merging contrastive learning loss from view enhancement with the BPR loss function:

$$\mathcal{L}_{all} = \alpha\mathcal{L}_{brp} + \beta\left(\mathcal{L}_{c^u} + \mathcal{L}_{c^i}\right) + \gamma\,\|\Theta\|^2 \tag{18}$$

5. Evaluation

In this section, we evaluates the effectiveness of the HCFCL model and examines the impact of its key components. Experiments are designed to address the following questions:

- **RQ1**: How does the performance of the HCFCL model compare with other models?
- **RQ2**: What impact do the key components of the HCFCL model have on its performance?
- **RQ3**: How does the HCFCL model cope with data sparsity issue?
- **RQ4**: What influence do hyperparameters have on the performance of the HCFCL model?

5.1. Experimental settings

5.1.1. Evaluation datasets

To comprehensively evaluate the HCFCL model's performance, we use three publicly available datasets derived from real-world scenarios for testing. Table 1 shows the analysis of datasets. The following is the detailed descriptions of each dataset:

- **Yelp**: The dataset originates from the user review platform Yelp. It includes a wealth of data related to users and items (e.g. business attributes, user reviews, areas), making it widely used in research on recommendation systems.
- **Epinions and Ciao**: These two datasets originate from consumer review websites, where users' trust relationships form a social network. The datasets predominantly consist of user evaluations of items and the trust relationships between users.

5.1.2. Evaluation metrics

The experiment samples one item with which the user has an interaction as a positive sample and 99 items that the user has not interacted with as negative samples for each user to effectively evaluate the model.

In this experiment, two widely used evaluation metrics in recommendation system research are adopted to measure the model's performance:

- **HR**: This metric aims to measure the accuracy of the recommendation system, specifically represented by the proportion of positive items in the recommendation list.

$$HR = \frac{\sum_{i=1}^{S} hit\,(i)}{S} \tag{19}$$

where S represents the total number of items the user has interacted with. The function $hit(i)$ is used to determine whether the ith item appears in the recommendation list. if the ith item is included, $hit(i)$ is assigned a value of 1; otherwise, $hit(i) = 0$.

- **NDCG**: It evaluates the results of recommendation by considering the position ranking of positive items in the recommendation list.

$$NDCG = \frac{1}{S}\sum_{i=1}^{S} \frac{2^{hit(i)} - 1}{\log_2\,(p+1)} \tag{20}$$

p is the position ranking of the positive item in the recommendation list.

Here, we adopt top-10 value on the two metric to evaluate the model.

5.1.3. Baselines

The HCFCL model compares with different baselines to verify its effectiveness. The detailed description of various recommendation system baselines is as follows:

Graph Neural Network-based Recommendation

- LightGCN [13]: Modeling user–item interaction data through graph convolutional layers without nonlinear projections to extract implicit features of users and items.
- DGRec [43]: Mining user dynamic interests and context-related social information through dynamic graph attention network.
- NGCF [7]: Stacking multiple graph message passing layers to inject latent collaborative filtering with high-order connectivity into the embedding.
- KGAT [14]: The information of the knowledge graph entity related to the bipartite graph is integrated into the nodes' embeddings by a knowledge-aware attention mechanism, achieving information screening of high-order relationships.

Heterogeneous Graph-based Recommendation

- DANSER [44]: Using dual GAT to learn dual social effects in user domain and item domain respectively
- HAN [45]: the information of different neighbor nodes under the same meta-path and the information of different semantic nodes in different meta-paths are learned by the attention mechanism respectively.
- HGT [46]: Utilizing edge-type and node-type dependent attention mechanisms to refine the type-specific representations of the heterogeneous graph nodes.
- GraphRec [47]: Integrating social relations and user–item interactions using a separate aggregation technique to learn latent factors of user and item.
- HeCo [48]: The node embedding of the two views of local structure and high-order multi-hop structure are learned respectively by the GNN. Then, contrastive learning is used to supervise the two views collaboratively.
- SMIN [49]: Acquire item-side knowledge information and high-level semantic relationships in heterogeneous graphs, and perform joint training through self-supervised mutual information learning.
- HGCL [50]: Transforming the information of heterogeneous side to enrich representations in the paradigm of graph contrastive learning.

Table 2
Comparison for different models across the three datasets.

Models	Yelp		Epinions		Ciao	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
Graph Neural Network-based Recommendation						
LightGCN	–	–	0.7845	0.6120	0.6597	0.4631
DGRec	0.7950	0.5593	0.7603	0.5668	0.6653	0.4953
NGCF	0.8265	0.5854	0.7984	0.5945	0.6945	0.4894
KGAT	0.7881	0.5501	0.7510	0.5578	0.6601	0.4512
Heterogeneous Graph-based Recommendation						
DANSER	0.8077	0.5692	0.7714	0.5741	0.6730	0.4521
HAN	0.7731	0.5604	0.7505	0.5275	0.6589	0.4469
HGT	0.8364	0.5883	0.8150	0.6126	0.6939	0.4869
GraphRec	0.8098	0.5679	0.7723	0.5751	0.6825	0.4730
HeCo	0.8359	0.5847	0.7998	0.5910	0.6867	0.4867
SMIN	0.8478	0.5993	0.8179	0.6137	0.7108	0.5012
HGCL	**0.8712**	0.6310	0.8367	0.6413	0.7376	0.5261
Hypergraph-based Recommendation						
MHCN	0.8344	0.5799	0.8201	0.6158	0.7053	0.4928
HCCF	0.8604	0.5938	0.8291	0.6268	0.7072	0.5092
DHCF	0.8063	0.5161	0.8063	0.5761	–	–
HCFCL	0.8655	**0.6356**	**0.8441**	**0.6503**	**0.7391**	**0.5315**

Hypergraph-based Recommendation

- MHCN [18]: Learning high-level social relationships using multi-channel hypergraph convolution.
- HCCF [40]: Based on the original interaction graph encoder, a parameterized hypergraph structure is adopted to capture local and global collaborative relationships.
- DHCF [19]: Dividing the bipartite graph into user and item channels for semantic distinction, and the high-order correlation between nodes is mined through jump hypergraph convolution in each channel.

5.2. Performance comparison (RQ1)

Table 2 presents the results of the HCFCL model compared to other benchmark recommendation system models with the performance on three datasets. From the evaluation results, the conclusions are summarized as follows:

- The HCFCL model demonstrates significant performance improvements on both HR and NDCG metrics compared to other models on the Yelp, Epinions, and Ciao datasets. The results confirm the effectiveness of the HCFCL model. We attribute this success to: (1) The embedding generative hypergraph network in the HCFCL model effectively mines the latent collaborative signals in the interaction data; (2) The feature knowledge personalized transfer network enables the HCFCL model to transfer the group common feature information.
- In Graph Neural Network-based Recommendation, except for NGCF, the other methods mainly capture the explicit collaborative signals hidden in the neighboring nodes, but the performance is not as good as NGCF. This indicates that a significant amount of valuable information is hidden within user–item interaction data, which can enhance recommendation performance. It also shows that it is necessary for the HCFCL model to mine implicit collaborative signals.
- In Heterogeneous Graph-based Recommendation, methods that divide data into different views to process according to their characteristics (e.g. GraphRec, HeCo, SMIN, and HGCL) perform better than mixed processing methods (e.g. DANSER, HAN). This highlights the existence of diverse and intricate types of data relationships within the interaction data. Each relationship type contains distinct semantic information, such as social semantics between users and interaction semantics between users and items. Dividing this data into multiple views based on different relationship types can alleviate semantic conflicts. The superior performance of the HCFCL model is not only due to multi-view division mitigating the semantic conflicts of different relationship types but also because it can adopt different operations based on the characteristics of data from various relationship types, capturing complex semantic information in detail. That is, the combination of embedding generative hypergraph network and GCN can better mine the semantic information in heterogeneous collaborative information in interaction data. Additionally, we noticed that among these methods that divide the processing into multiple views, only GraphRec did not utilize self-supervised learning. Consequently, its performance was inferior to the other models that did. It shows that self-supervised learning methods can promote the performance improvement of multi-view processing methods.
- In the Hypergraph-based Recommendation approach, the hypergraph is employed to capture the high-order correlations among entities in the interaction data. Compared with the GNN-based Recommendation method, although there is a significant improvement, the difference between high-order correlation and local correlation is ignored. In HCFCL, the explicit collaborative signal represents the correlation of the local subgraph, and the implicit collaborative signal represents the high-order correlation. The difference between these two heterogeneous collaborative signals is alleviated by the Feature Knowledge Personalized Transfer Network, and the results strongly confirm the excellent performance and advantages of the HCFCL model.

5.3. Ablation study of HCFCL (RQ2)

To verify the effectiveness of mining heterogeneous collaborative signals for recommendation, we conducted relevant ablation experiments on the HCFCL model:

- w/o-hyper: Model HCFCL without the embedding generative hypergraph network means that it cannot mine the potential collaborative signals hidden in the interaction data.
- w/o-trans: Model HCFCL without feature knowledge personalized transfer network means that it cannot customize the group common feature information contained in the mined potential collaborative signals.
- w/o-cl: Modeling HCFCL without contrastive learning means that it is unable to capture the consistency of heterogeneous collaborative signals.

Table 3
Ablation experiment about the key components of HCFCL.

Variants	Yelp		Epinions		Ciao	
Metrics	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
w/o-hyper	0.8531	0.6248	0.8308	0.6478	0.7191	0.5253
w/o-trans	0.8505	0.6257	0.8331	0.6493	0.7242	0.5228
w/o-cl	0.8481	0.6046	0.8318	0.6356	0.7304	0.5263
HCFCL	0.8655	0.6356	0.8441	0.6503	0.7391	0.5315

Table 4
The user numbers with different interaction numbers range in the datasets.

Interaction	Yelp	Epinions	Ciao
0–20	151 926	8032	3384
20–40	5384	3551	1391
40–60	1739	1148	560
60–80	791	500	299
80–100	368	286	169

Table 3 shows the results of the ablation experiments, which clearly demonstrate the significantly superior performance of the complete HCFCL model compared to its various simplified variants on the recommendation task. When contrastive learning is removed, the w/o-cl variant shows a decrease in HR@10 and NDCG@10 metrics in the three datasets compared to the full HCFCL model. This demonstrates that contrastive learning effectively captures the consistency of heterogeneous collaborative signals, achieving deep fusion of these signals and alleviating the data sparsity issue.

The superior performance of HCFCL compared to the w/o-trans variant validates the effectiveness of the feature knowledge personalized transfer network. This network filters out irrelevant information from the group common feature information contained in the latent collaborative signals and transforms it based on user characteristics.

The w/o-hyper variant performs worse than the HCFCL model because it lacks the deep mining of interaction data by the embedding generative hypergraph network. It emphasizes the importance of deeply mining latent collaborative signals in interaction data. The extraction of implicit collaborative signals inevitably introduces noise to some extent. First, it originates from the inherent noise in the interaction data. When explicit collaborative signals are captured, this noise is propagated through the message-passing mechanism in the graph encoder. Since the mining of implicit collaborative signals relies on user and item embeddings derived from explicit collaborative signals, the noise embedded in these representations is further amplified during the hypergraph construction and learning process. Second, the hypergraph structure's deviation from group construction introduces additional noise. However, when the interaction data is sparse, the information gain provided by the mining of implicit collaborative signals outweighs the negative impact of the noise, making this method beneficial overall. This can be demonstrated in the ablation experiments of the w/o-hyper variant.

5.4. HCFCL in alleviating data sparsity issue (RQ3)

To validate the ability of the HCFCL model to cope with the data sparsity issue within recommendation systems, it is compared with other baselines on selected sparse datasets. As shown in Table 4, Users across the three datasets who have fewer than 100 interactions are categorized into five groups. These groups contain progressively fewer users as the interaction count increases. The experimental results in Fig. 4 reveal that the performance of the HCFCL model shows a high degree of stability in the face of changes in interaction, without significant fluctuations. Additionally, HCFCL shows superior recommendation effects compared to the HGCL and HCCF models in all subdivided dataset groups, further confirming its excellent ability to handle sparse data.

Table 5
Comparison for different models across the MOOCs datasets.

Models	MOOCCourse		MOOCCube	
	HR@10	NDCG@10	HR@10	NDCG@10
BPR	0.145	0.043	0.095	0.030
BPR-HFT	0.156	0.053	0.099	0.038
LightGCN	0.171	0.070	0.109	0.051
RuleRec	0.179	0.074	0.114	0.059
TP-GNN	0.192	0.085	0.156	0.066
PGPR	0.229	0.090	0.200	0.080
ADAC	0.239	0.103	0.208	0.083
KRRL	0.263	0.121	0.217	0.094
HCFCL	**0.304**	**0.162**	**0.246**	**0.129**

The HCFCL model demonstrates outstanding performance in mitigating the data sparsity issue, which can be attributed to the extraction of implicit collaborative signals that compensate for the scarce information from explicit collaborative signals.

5.5. Parametric study (RQ4)

This section examines the impact of various settings of the following important hyperparameters on the HCFCL model: Graph layer number l, hyperedge number g, and node representation dimensionality d. Fig. 5 presents the experimental results, where only one hyperparameter was adjusted for each individual experiment, while the other parameters were kept at their preset default values.

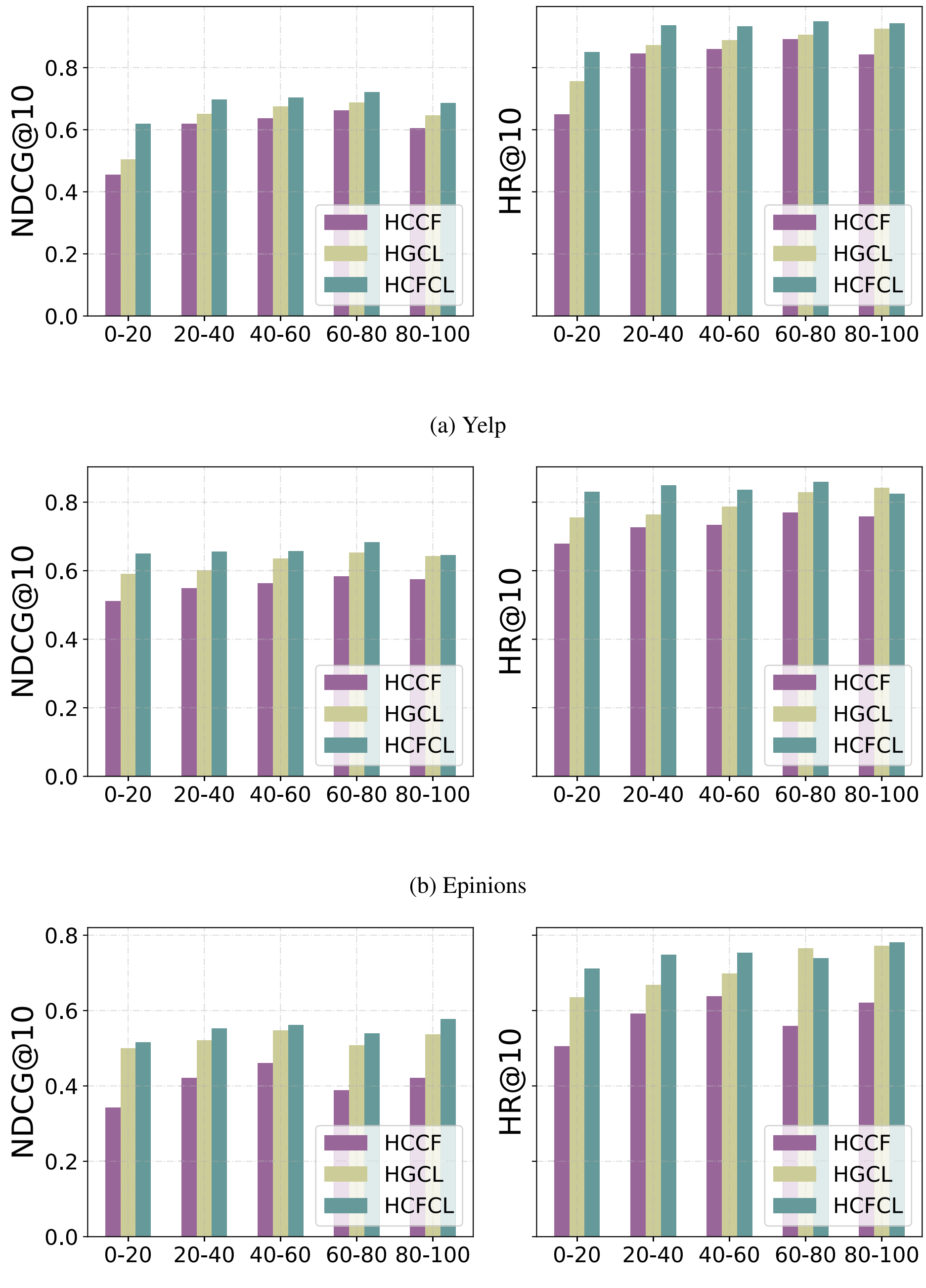
The numbers of the graph layer l. The model performs optimally when the number of graph layers $l = 2$. A single graph layer may not fully leverage collaborative signals within interaction data, while higher numbers of GNN layers can lead to oversmoothing issues, resulting in nodes' representations becoming more homogeneous.

Hyperedge number g. When the number of hypergraphs $g = 256$, the HCFCL model achieves optimal performance. An excessive number of groups increases the difficulty in determining the relationships between nodes and groups, making it challenging for the generative hyperedge incidence matrix to accurately describe the associations between nodes and groups. Conversely, too few hyperedges diminishes the prominence of the group common features, hindering effective transfer of the group common feature information.

Node representation dimensionality d. We noticed that the model's performance initially improves and then decreases as the node representation dimensionality increases, reaching the highest point when Node representation dimensionality $d = 32$. This shows that the amount of information carried by Node representation increases with the increase of dimensionality. When d exceeds 32, the model will be overfitted, thus degrading the recommendation performance.

5.6. Model generalization analysis

To test the generalization ability of the HCFCL model beyond the domain of social recommendation, we conducted experiments on the model in the e-learning scenario and compared it with other models. The experiment was based on the real-world MOOCCourse and MOOCCube datasets. MOOCCourse consists of 1302 courses from 23 categories and 82,535 students. These 1302 courses involve 27,173 concepts and 411 prerequisites. In MOOCCourse, there are only 458,453 student-course interactions, with each student taking an average of approximately 5.55 courses. In the MOOCCube dataset, 55,203 students



(a) Yelp

(b) Epinions

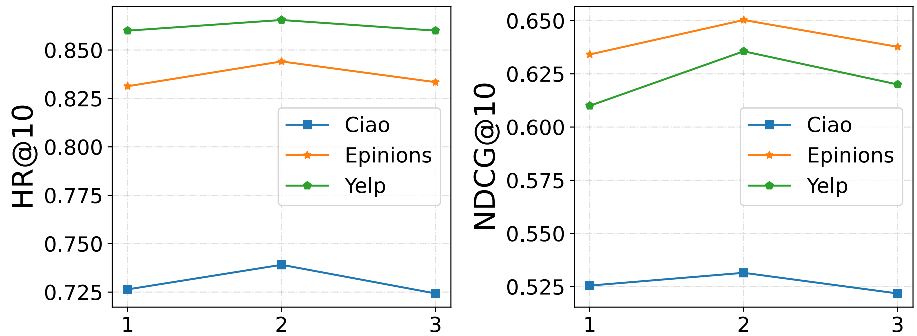
(c) Ciao

Fig. 4. Performance comparison under sparse data with different interaction range.

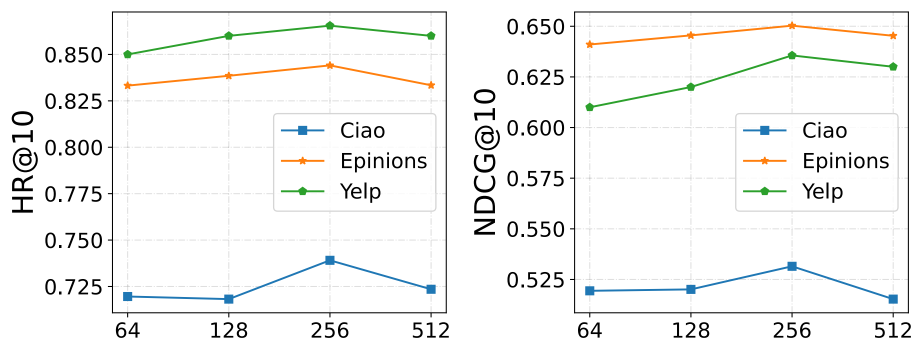
interact with 706 courses, resulting in 354,541 student-course interactions, with each student taking an average of about 6.42 courses. The courses from 20 categories in MOOCCube involve 23,207 concepts and 352 prerequisites. From the perspective of the average number of courses taken by students, both datasets are data-sparse.

The baseline models for comparison are divided into two main categories. One category focuses on user and course representation learning using student-course interactions, including models like BPR [51], BPR-HFT [52], LightGCN [13], and TP-GNN [53]. The other category introduces domain knowledge via knowledge graphs, including models like RuleRec [54], PGPR [55], ADAC [56], and KRRL [57].

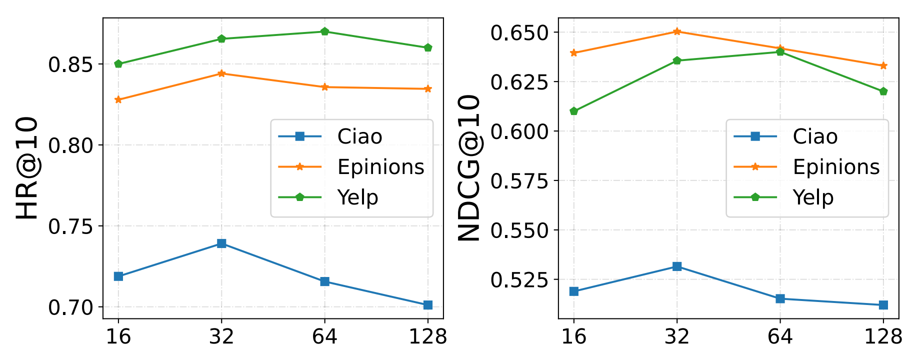
As shown in Table 5, the HCFCL model showed a significant performance improvement over other models on both MOOCCourse and MOOCCube datasets. Compared to methods that only use student-course interactions, HCFCL is better at mining students' preferences and course features from the interaction data, thanks to the exploration of heterogeneous collaborative signals, especially in data-sparse scenarios. Most of the methods that introduce domain knowledge of knowledge graphs are better than the methods that only use student-course interactions. Domain knowledge can make up for the problem of insufficient data caused by data sparsity. However, these methods are less effective than the HCFCL model, indicating that deep mining of interaction data is still a direct and effective way to achieve excellent recommendation results. The effectiveness of domain knowledge introduction is influenced by multiple factors, such as the quality of the domain knowledge content, the relevance of knowledge to the corresponding items, and how the knowledge is processed. Methods like RuleRec [54],



(a) Graph layer l



(b) Hyperedge number g



(c) Node representation dimensionality d

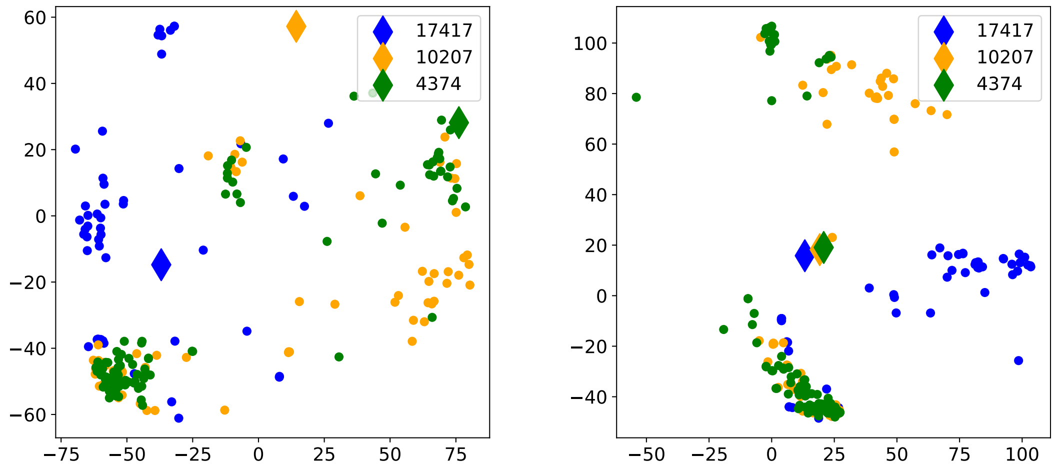
Fig. 5. Hyperparameter research of the HCFCL.

PGPR [55], ADAC [56], and KRRL [57] are affected by the quality of knowledge graph construction, and poor-quality knowledge graphs can even degrade recommendation performance. In addition, different domains have different knowledge, specific operations are needed to handle the relevant knowledge for each domain, which limits the generalization ability of domain-knowledge-based approaches compared to interaction-based recommendation methods. The HCFCL model not only performs well in social recommendation but also outperforms domain-knowledge-based methods in the e-learning scenario, proving its generalization ability to handle recommendation problems across different domains.

5.7. Model complexity analysis

We measured the time (in seconds) of each batch and epoch during training for HCFCL and other models. Our experiments were conducted on a Linux machine equipped with a GeForce RTX 3090 GPU. As shown in Table 6, the time taken by the HCFCL model and the HCCF model on the three datasets is not much different, but much lower than the time consumed by the HGCL model. This shows that the HCFCL model not only has excellent performance but also has a lower time complexity. The explicit collaborative signals capturing part takes $\mathcal{O}\left(|A_{He}| \times l \times d\right)$, where $|A_{He}|$ denotes the edge number in the adjacency matrix. The time complexity for the embedding generative hypergraph network is $\mathcal{O}((q+s) \times (g+d) \times d \times l \times g)$. Additionally, the main complexity of the feature knowledge personalized transfer network concentrates on knowledge extraction, with a time complexity of $\mathcal{O}\left((q+s) \times d^2\right)$.

The time consumption of the HCFCL model varies under different scenarios. In data-sparse conditions, where the number of nodes in the user–item bipartite graph typically exceeds the number of edges, the time consumption of the model is primarily concentrated in the



(a) HCCF (b) HCFCL

Fig. 6. Visualization of representations under dimensionality reduction of the t-SNE algorithm.

Table 6
The running time per batch and per epoch for different models on three datasets.

Model	Yelp		Epinions		Ciao	
	Batch	Epoch	Batch	Epoch	Batch	Epoch
HGCL	2.650 s	321.563 s	3.900 s	308.332 s	3.318 s	112.977 s
HCCF	0.074 s	19.346 s	0.057 s	11.505 s	0.040 s	4.145 s
HCFCL	0.072 s	19.306 s	0.064 s	12.404 s	0.051 s	5.057 s

embedding generative hypergraph network. In this case, the time cost is mainly influenced by the number of nodes and the dimensionality of node representations. When the dataset is large, and the number of interactions significantly exceeds that of users and items, the model's time consumption shifts toward the explicit collaborative signals capturing, where the time cost is primarily affected by the number of interactions (edges in the bipartite graph) and node representations. Under data-sparse scenarios, the HCFCL model's time consumption is higher compared to the lightweight recommendation model LightGCN [13]. However, as the volume of interaction data increases, the time consumption of HCFCL gradually approaches that of LightGCN [13]. Therefore, the HCFCL model improves recommendation quality at the cost of increased time consumption in low-interaction situations, encouraging users to generate more high-quality interaction data. As interaction data continues to grow, the incremental time cost of HCFCL gradually slows down while maintaining relatively high recommendation quality.

6. Case study

To intuitively demonstrate the effectiveness of the model in learning representations for users and items, we selected three users with sparse interactions along with their corresponding items for visualization of representations. Fig. 6 shows the visualization result of the representation dimensionality reduction using the s-tne algorithm, where diamonds represent users and dots represent items.

In comparison with Fig. 6(a), Fig. 6(b) shows that the items of the same color are more tightly clustered, with a noticeable spatial separation between items of different colors. This indicates that the HCFCL model is more effective at mining user preferences, not only identifying them but also clearly distinguishing between different preferences. It groups items with similar characteristics that cater to specific user preferences. This is primarily due to the personalized transfer of group common feature information within the HCFCL model, which both emphasizes user preferences and item characteristics while compensating for data scarcity. On the other hand, the HCCF model introduces a hypergraph to capture high-order dependencies across the entire graph, building upon collaborative signals mined from interactive data via graph convolutional networks. This allows the model to gather neighbor node information with high-order connectivity, which helps mitigate the impact of data sparsity. However, the HCCF model does not differentiate the neighbor node information with high connectivity according to the characteristics of the source node. As a result, nodes connected by the same hyperedge aggregate similar information, leading to a blurring of user preferences and item characteristics. In Fig. 6(a), the items of the same color are dispersed in the space.

In addition, Fig. 6(b) shows a clear separation between users and items. This is the result of the HCFCL model treating users and items separately when mining implicit collaborative signals, which enables it to better distinguish between users and items. The HCCF model constructs a hypergraph for both user and item nodes, with a hyperedge connecting the user and item nodes. This makes the user and item nodes in the HCCF model mixed in space and difficult to distinguish.

7. Conclusion

This paper primarily aims at mining heterogeneous collaborative signals in recommendation systems. To this end, we propose the HCFCL model. It deeply mines the implicit collaborative signals in interaction data by the embedding generative hypergraph network to supplement the scarcity of explicit collaborative signals in the data sparsity issue. In order to alleviate the difference between the group common feature and individual characteristics, we designed a personalized bridge function, which, with the assistance of the meta network, transfers the group common feature information for user/item personalization. Finally, through contrastive learning, the HCFCL model captures the consistency of heterogeneous collaborative signals to enhance embedding learning. The HCFCL model is verified with other methods on the three publicly available datasets. Experiments conducted on Yelp, Epinions, and Ciao have demonstrated the HCFCL model's superior recommendation performance. Supernumerary experiments and case study prove that the HCFCL model can address the data sparsity issue better than other methods.

In this work, the HCFCL model processes the interaction data from the structural perspective, in order to enable the model to meet the recommendation tasks in most scenarios, especially when there is only sparse interaction data. However, the HCFCL model has certain scalability and can integrate relevant domain knowledge or side information in social recommendation. Using appropriate operations to replace the normal distribution to randomly obtain the initial representation of users and items for side information processing can assist model in mining interaction data based on the understanding of side information, thereby reducing the difficulty of mining heterogeneous collaborative signals. For example, Side information, such as item attributes and social connections, can be used to construct corresponding graph structures. A graph encoder can then encode both user and item nodes, capturing the relationships and dependencies within the graph to enhance the representation learning process. The side information representation obtained by encoding can not only serve as the initial user and item representation for capturing explicit collaborative signals of the model, but also as a source of feature knowledge for Feature Knowledge Extraction, assisting in the personalized transfer of group common feature information.

8. Future work

In social recommendation scenarios, side information such as social connections or item attributes plays a crucial role. Leveraging this side information to enrich representations, and integrating it into a multi-task learning framework through auxiliary tasks, is a promising direction for joint optimization. This approach holds great potential for improving recommendation accuracy and capturing more nuanced patterns in the data. In addition to interaction data, there is also abundant multimodal information related to users and items. In the current work, only the single graph structure modal information of interaction data is involved. How to extract information in multimodal data and alleviate the modal conflict of multimodal information is a direction we need to explore.

CRediT authorship contribution statement

Chaojun Meng: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Changfan Pan:** Supervision. **Hongji Shu:** Supervision. **Qing Wang:** Supervision. **Hanghui Guo:** Supervision. **Jia Zhu:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Zhejiang Provincial Natural Science Foundation of China under Grant No. LY23F020010, the National Natural Science Foundation of China under Grant (No. 62337001, No. 62077015), and the National Key R&D Program of China under Grant No. 2022YFC3303600.

Data availability

Data will be made available on request.

References

[1] J. Gao, S. Han, H. Zhu, S. Yang, Y. Jiang, J. Xu, B. Zheng, Rec4ad: A free lunch to mitigate sample selection bias for ads CTR prediction in Taobao, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023, pp. 4574–4580.

[2] R. Kuo, S.-S. Li, Applying particle swarm optimization algorithm-based collaborative filtering recommender system considering rating and review, Appl. Soft Comput. 135 (2023) 110038.

[3] R. Xie, Q. Liu, L. Wang, S. Liu, B. Zhang, L. Lin, Contrastive cross-domain recommendation in matching, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 4226–4236.

[4] Z. Yi, X. Wang, I. Ounis, C. Macdonald, Multi-modal graph contrastive learning for micro-video recommendation, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 1807–1811.

[5] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.

[6] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 173–182.

[7] X. Wang, X. He, M. Wang, F. Feng, T.-S. Chua, Neural graph collaborative filtering, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 165–174.

[8] K. Li, C.-D. Wang, J.-H. Lai, H. Yuan, Self-supervised group graph collaborative filtering for group recommendation, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2023, pp. 69–77.

[9] Y. Wang, C. Li, M. Li, W. Jin, Y. Liu, H. Sun, X. Xie, J. Tang, Localized graph collaborative filtering, in: Proceedings of the 2022 SIAM International Conference on Data Mining, SDM, SIAM, 2022, pp. 540–548.

[10] Z. Lin, C. Tian, Y. Hou, W.X. Zhao, Improving graph collaborative filtering with neighborhood-enriched contrastive learning, in: Proceedings of the ACM Web Conference 2022, 2022, pp. 2320–2329.

[11] J. Wang, J. Ren, Graphormer based contrastive learning for recommendation, Appl. Soft Comput. 159 (2024) 111626.

[12] L. Yang, S. Wang, Y. Tao, J. Sun, X. Liu, P.S. Yu, T. Wang, Dgrec: Graph neural network for recommendation with diversified embedding generation, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2023, pp. 661–669.

[13] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, 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, 2020, pp. 639–648.

[14] X. Wang, X. He, Y. Cao, M. Liu, T.-S. Chua, Kgat: Knowledge graph attention network for recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 950–958.

[15] R. Ying, R. He, K. Chen, P. Eksombatchai, W.L. Hamilton, J. Leskovec, Graph convolutional neural networks for web-scale recommender systems, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 974–983.

[16] L. Xia, Y. Xu, C. Huang, P. Dai, L. Bo, Graph meta network for multi-behavior recommendation, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 757–766.

[17] Z. Fan, K. Xu, Z. Dong, H. Peng, J. Zhang, P.S. Yu, Graph collaborative signals denoising and augmentation for recommendation, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023, pp. 2037–2041.

[18] J. Yu, H. Yin, J. Li, Q. Wang, N.Q.V. Hung, X. Zhang, Self-supervised multi-channel hypergraph convolutional network for social recommendation, in: Proceedings of the Web Conference 2021, 2021, pp. 413–424.

[19] S. Ji, Y. Feng, R. Ji, X. Zhao, W. Tang, Y. Gao, Dual channel hypergraph collaborative filtering, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 2020–2029.

[20] J. Liao, W. Zhou, F. Luo, J. Wen, M. Gao, X. Li, J. Zeng, Sociallgn: Light graph convolution network for social recommendation, Inform. Sci. 589 (2022) 595–607.

[21] Y. Chen, Z. Liu, J. Li, J. McAuley, C. Xiong, Intent contrastive learning for sequential recommendation, in: Proceedings of the ACM Web Conference 2022, 2022, pp. 2172–2182.

[22] S. Fan, J. Zhu, X. Han, C. Shi, L. Hu, B. Ma, Y. Li, Metapath-guided heterogeneous graph neural network for intent recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2478–2486.

[23] C. Chen, W. Ma, M. Zhang, Z. Wang, X. He, C. Wang, Y. Liu, S. Ma, Graph heterogeneous multi-relational recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 3958–3966.

[24] T. Munkhdalai, H. Yu, Meta networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 2554–2563.

[25] J. Gordon, J. Bronskill, M. Bauer, S. Nowozin, R.E. Turner, Meta-learning probabilistic inference for prediction, 2018, arXiv preprint arXiv:1805.09921.

[26] E. Grefenstette, B. Amos, D. Yarats, P.M. Htut, A. Molchanov, F. Meier, D. Kiela, K. Cho, S. Chintala, Generalized inner loop meta-learning, 2019, arXiv preprint arXiv:1910.01727.

[27] L. Ding, P. Liu, W. Shen, W. Lu, S. Chen, Gradient-based meta-learning using uncertainty to weigh loss for few-shot learning, 2022, arXiv preprint arXiv:2208.08135.

[28] S.W. Yoon, J. Seo, J. Moon, Tapnet: Neural network augmented with task-adaptive projection for few-shot learning, in: International Conference on Machine Learning, PMLR, 2019, pp. 7115–7123.
[29] V. Garcia, J. Bruna, Few-shot learning with graph neural networks, 2017, arXiv preprint arXiv:1711.04043.
[30] Z. Chu, H. Wang, Y. Xiao, B. Long, L. Wu, Meta policy learning for cold-start conversational recommendation, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2023, pp. 222–230.
[31] Y. Du, X. Zhu, L. Chen, Z. Fang, Y. Gao, Metakg: Meta-learning on knowledge graph for cold-start recommendation, IEEE Trans. Knowl. Data Eng. (2022).
[32] R. Guan, H. Pang, F. Giunchiglia, Y. Liang, X. Feng, Cross-domain meta-learner for cold-start recommendation, IEEE Trans. Knowl. Data Eng. (2022).
[33] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, Q.V.H. Nguyen, Are graph augmentations necessary? simple graph contrastive learning for recommendation, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 1294–1303.
[34] R. Qiu, Z. Huang, H. Yin, Z. Wang, Contrastive learning for representation degeneration problem in sequential recommendation, in: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022, pp. 813–823.
[35] X. Cai, C. Huang, L. Xia, X. Ren, Lightgcl: Simple yet effective graph contrastive learning for recommendation, 2023, arXiv preprint arXiv:2302.08191.
[36] J. Yu, X. Xia, T. Chen, L. Cui, N.Q.V. Hung, H. Yin, Xsimgcl: Towards extremely simple graph contrastive learning for recommendation, IEEE Trans. Knowl. Data Eng. (2023).
[37] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, X. Xie, Self-supervised graph learning for recommendation, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 726–735.
[38] Y. Yang, C. Huang, L. Xia, C. Li, Knowledge graph contrastive learning for recommendation, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 1434–1443.
[39] D. Cai, S. Qian, Q. Fang, J. Hu, W. Ding, C. Xu, Heterogeneous graph contrastive learning network for personalized micro-video recommendation, IEEE Trans. Multimed. (2022).
[40] L. Xia, C. Huang, Y. Xu, J. Zhao, D. Yin, J. Huang, Hypergraph contrastive collaborative filtering, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 70–79.
[41] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, 2014, arXiv preprint arXiv:1406.1078.
[42] Y. Zhu, Z. Tang, Y. Liu, F. Zhuang, R. Xie, X. Zhang, L. Lin, Q. He, Personalized transfer of user preferences for cross-domain recommendation, in: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022, pp. 1507–1515.
[43] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, J. Tang, Session-based social recommendation via dynamic graph attention networks, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019, pp. 555–563.
[44] Q. Wu, H. Zhang, X. Gao, P. He, P. Weng, H. Gao, G. Chen, Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems, in: The World Wide Web Conference, 2019, pp. 2091–2102.
[45] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, P.S. Yu, Heterogeneous graph attention network, in: The World Wide Web Conference, 2019, pp. 2022–2032.
[46] Z. Hu, Y. Dong, K. Wang, Y. Sun, Heterogeneous graph transformer, in: Proceedings of the Web Conference 2020, 2020, pp. 2704–2710.
[47] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, D. Yin, Graph neural networks for social recommendation, in: The World Wide Web Conference, 2019, pp. 417–426.
[48] X. Wang, N. Liu, H. Han, C. Shi, Self-supervised heterogeneous graph neural network with co-contrastive learning, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 1726–1736.
[49] X. Long, C. Huang, Y. Xu, H. Xu, P. Dai, L. Xia, L. Bo, Social recommendation with self-supervised metagraph informax network, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 1160–1169.
[50] M. Chen, C. Huang, L. Xia, W. Wei, Y. Xu, R. Luo, Heterogeneous graph contrastive learning for recommendation, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2023, pp. 544–552.
[51] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, 2012, arXiv preprint arXiv:1205.2618.
[52] J. McAuley, J. Leskovec, Hidden factors and hidden topics: understanding rating dimensions with review text, in: Proceedings of the 7th ACM Conference on Recommender Systems, 2013, pp. 165–172.
[53] J. Wang, H. Xie, F.L. Wang, L.-K. Lee, O.T.S. Au, Top-N personalized recommendation with graph neural networks in MOOCs, Comput. Educ.: Artif. Intell. 2 (2021) 100010.
[54] W. Ma, M. Zhang, Y. Cao, W. Jin, C. Wang, Y. Liu, S. Ma, X. Ren, Jointly learning explainable rules for recommendation with knowledge graph, in: The World Wide Web Conference, 2019, pp. 1210–1221.
[55] Y. Xian, Z. Fu, S. Muthukrishnan, G. De Melo, Y. Zhang, Reinforcement knowledge graph reasoning for explainable recommendation, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 285–294.
[56] K. Zhao, X. Wang, Y. Zhang, L. Zhao, Z. Liu, C. Xing, X. Xie, Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 239–248.
[57] Y. Lin, W. Zhang, F. Lin, W. Zeng, X. Zhou, P. Wu, Knowledge-aware reasoning with self-supervised reinforcement learning for explainable recommendation in MOOCs, Neural Comput. Appl. 36 (8) (2024) 4115–4132.