IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 January 2024, accepted 25 April 2024, date of publication 8 May 2024, date of current version 24 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3398367

RESEARCH ARTICLE

A Recommendation Method Based on Multi-Source Heterogeneous Hypergraphs and Contrastive Learning

SHANSHAN WAN AND JIAQI DING
School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China
Corresponding author: Jiaqi Ding (2108110021004@stu.bucea.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61902016.

ABSTRACT Fusion of multi-source information is one of the primary methods to alleviate data sparsity in recommender systems (RS). Hypergraphs have shown remarkable capabilities in dealing with the diversity of multi-source information, especially in modeling high-order user-item relationships. However, most hypergraph research focuses on constructing hypergraphs using a particular type of hyperedge, which might not fully capture the high-order implicit associations among heterogeneous information sources. Furthermore, existing Hypergraph Attention Networks (Hyper-GAT), mainly emphasizes on information propagation between nodes and hyperedges, insufficiently exploring density information within complex hyperedges. Moreover, when hypergraphs combine data from multiple heterogeneous graphs, redundant information from different viewpoints can reduce the effectiveness of multivariate modeling hyperedges. Thus, we propose a recommendation algorithm based on multi-source heterogeneous hypergraphs and contrastive learning (MHCLR), which improves recommendation accuracy through multi-source information fusion and higher-order information correlation. First, multi-source heterogeneous hypergraphs (MHC) are generated by combining distance, behavior, attribute, and prediction hyperedges. We mine associations among multi-source information, enhancing high-order semantic connections among users and items. Then, a Spatial Density Hypergraph Attention Network (SD-HGAT) is proposed based on composite hyperedges, which enriches the user and item embedding representations by focusing on nodes and hyperedges density. Finally, we design a multiple cross view contrastive learning (MCC) that compares views centered around knowledge graphs with hypergraphs, improving the accuracy of multivariate relationship modeling and enabling multi-level user profiling construction. It is observed that MHCLR outperforms the baselines in terms of Recall, Precision, and NDCG based on the experimental results by Yelp, Last-FM, and Douban.

INDEX TERMS Multi-source heterogeneous hypergraph, hypergraph attention network, contrastive learning, recommender system.

I. INTRODUCTION

The digital age has led to an explosion of information, causing information overload and making it hard to meet users' personalized needs. Recommender systems help users quickly find relevant information from vast data. Currently, recommender systems have broad applications in various domains such as session-based recommendation [1], [2], product recommendation [3], news recommendation [4], music recommendation [5], and social recommendation [6].

However, recommender systems frequently encounter challenges, including data sparsity [7] and cold start problems [8]. Some algorithms like traditional collaborative filtering typically rely on user behavior data for recommendation [9]. Nevertheless, these algorithms often struggle to achieve satisfactory results due to data sparsity, in which users have no behavior data for most items, posing a challenge in accurately capturing user preferences. The idea of

The associate editor coordinating the review of this manuscript and approving it for publication was Yin Zhang.

multi-source information fusion has been introduced [10], [11], which adds more comprehensive information to tackle the issues arising from data sparsity.

Multi-source information typically encompasses various data types, including social network relationships, contextual information, as well as attributes data. Due to the progress in deep learning techniques, some studies have employed deep learning models, such as Recurrent Neural Networks (RNN), to extract high-level features from raw data to facilitate multi-source information fusion [12]. Additionally, there are studies that leverage knowledge from multiple sources for fused recommendation, offering domain knowledge and semantic associations [13]. Graph neural networks unify data from various sources into a single graph, capturing complex relationships and dependencies among multi-source information. Graph neural networks are extensively utilized in recommender systems, with various types of graphs including sequential graphs (Sd) [14], social network graphs (Sg) [15], heterogeneous information networks (HIN) [16], knowledge graphs (Kg) [17], and hypergraphs [18], [19]. Hypergraph effectively captures complex, high-order interactions among objects with multiple associations, overcoming information gaps in other graph models.

Hypergraph structures often rely on a single type of hyperedge, such as attributes or topics [20], [21]. When dealing with multi-source information, depending solely on a single-type hyperedge to capture complex user-item relationships within the multi-source data can lead to information loss. To address this issue, some studies have utilized multiple types of hyperedges to capture intricate associations. For instance, Gatta et al. [22] introduced the use of hypergraphs combined with machine learning on graphs in music recommendation. They designed hyperedges for user listening sequences, music topics, albums, and so on, to uncover potential associations between users and music. However, this approach may still lack comprehensiveness in capturing relationships between users and music, as it may not consider factors such as social relationships between users or the distance in audio features of music.

Hypergraph can establish many-to-many relationships and high-order relationships between nodes. To effectively capture these relationships, researchers have introduced the concept of hypergraph learning, which enables models to fully understand and utilize complex data associations [23]. Hypergraph neural networks (HGNN) [24] offer strong learning capabilities and model flexibility, contributing to overcoming computational complexities and the curse of high-dimensionality associated with hypergraph learning. Nevertheless, Yadati et al. [25] argued that HGNN models introduce excessive noise during information fusion, potentially affecting semi-supervised learning based on hypergraphs. To address this concern, they proposed the HyperGCN model, which filters out potential data noise to some extent. Additionally, Ding et al. [26] proposed the Hyper-GAT model, which combines graph attention networks and hypergraph neural networks to achieve greater expressive power in text representation learning while minimizing computational costs. The aforementioned studies primarily focus on information propagation between nodes and hyperedges in hypergraphs, capturing the structural information of hypergraphs in various ways. However, they overlooked some valuable implicit information. Liao et al. [27] introduced density as implicit information in hypergraph neural networks. They enhanced hypergraph information mining by calculating edge densities based on node densities, improving the results achieved in downstream tasks. Although the mentioned studies had integrated density information with nodes and hyperedges in hypergraph neural networks, they may not be suitable for capturing hidden connections within complex hyperedges involving multiple relationships or adequately learning density information among intricate hyperedges.

When integrating multi-source information in heterogeneous hypergraphs, it is necessary to consider the alignment between the same user or item from different information sources. Additionally, modeling complex relationships and dependencies among various variables on nodes and edges is difficult [28]. Multi-view contrastive learning is based on the principles of contrastive learning [29] and can address these issues. Through multi-view contrastive learning, information from different views can be compared and fused, resulting in more accurate data representations. This enhances feature representation and association learning in multivariable modeling tasks, helping recommender systems better model relationships between users and items. Moreover, it leverages complementary information from different views to overcome the limitations of a single view, leading to richer and more discriminative feature representations in multi-source information fusion. For instance, Chen et al. [30] proposed heterogeneous graph contrastive learning, which integrates semantic relationships from heterogeneous sources into user-item interaction modeling, enhancing knowledge transfer through contrastive learning across different views. Wu et al. [31] proposed the Multi-Behavior Multi-Perspective Contrastive Learning for recommendation framework. It simultaneously considers single-sequence views and global graph views in modeling multiple behaviors, capturing fine-grained differences between user behavior.

To address the challenges of modeling multi-variable relationships in multi-source heterogeneous information and mining complex density information within compound hyperedges, we present a method based on multi-source heterogeneous hypergraphs and contrastive learning. Specifically, we comprehensively fuse multi-source information, including user-item bipartite graphs (Bg), social network graphs, behavioral sequence graphs, and knowledge graphs, by leveraging compound hyperedges to generate multi-source heterogeneous hypergraphs. In the process of hypergraph learning representation, we incorporate density information with Hyper-GAT and propose the SD-HGAT. This network delves into the structural information intertwined between compound hyperedges and nodes, strengthening user and item representation learning while incorporating Gated Recurrent Unit (GRU) networks to capture users' short-term preferences. Additionally, we design a multi-view contrastive learning framework with the knowledge graph as the central component to serve as auxiliary training tasks. This helps alleviate errors in multivariate modeling within heterogeneous views, enabling a multi-faceted learning of user and item feature representations, thereby enhancing recommendation performance.

The main contributions of this study are as follows:

(1) A method is proposed that utilizes hypergraphs for multi-source information fusion. By employing compound hyperedges, we comprehensively generate multi-source heterogeneous hypergraphs. This approach facilitates the integration of diverse multi-source heterogeneous information, establishing hidden associations between data and enhancing the representation of high-order correlations among the data.

(2) We establish the SD-HGAT model, which incorporates density embedding into the hypergraph attention network. This model is combined with GRU networks to further explore implicit density structural information between compound hyperedges and nodes, thereby expanding the embedding space for users and items.

(3) Multiple cross views have been designed to facilitate the process of contrastive learning. By combining information across multiple views and contrasting it with the hypergraph-based information, we obtain more discriminative feature representations. This helps in reducing errors in multivariable modeling within the hypergraph, improving the accuracy of user-item feature representations, and enabling precise identification of user preferences and demands.

The paper is organized as follows: Section II discusses related research in multi-source fusion for recommenders, hypergraph construction and learning. Section III outlines our method for constructing a heterogeneous hypergraph and the MHCLR model, along with cross-view contrastive learning. Section IV provides experimental results, model comparisons, and ablation studies. Section V concludes the paper with contributions and future research suggestions.

II. RELATED WORK

This paper focuses on data sparsity, high-order user relationship modeling, and comprehensive user preference exploration. The current strategies to tackle these challenges include multi-source information fusion, such as utilizing hypergraphs to fuse multi-source information for modeling user-item relationships and feature extraction and using multi-view contrastive learning to mitigate data sparsity, enhance multivariable modeling accuracy. The following summarizes the related research.

A. MULTI-SOURCE INFORMATION FUSION IN RECOMMENDER SYSTEMS

Multi-source information fusion is currently a significant research direction in recommender systems. However, with the abundance of auxiliary information and the diversification of user demands, the recommendation performance of a single data source gradually becomes limited. Approaches for multi-source information fusion strive to integrate multiple data sources, combining rich auxiliary information such as geographical location, context, social networks, attributes, etc., into the recommendation task to enhance recommendation accuracy and personalization.

For example, Khelloufi et al. [32] improved context-aware service recommendation by leveraging social relationships among device owners. Sun [33] incorporated textual reviews, location-based Points of Interest (POI), and user latent factors to model deep user preferences in the sparse POI scenario. Yu [34] addressed issues in personalized travel recommendation, such as data sparsity and low recall, by mining context information in unstructured text, semantic and sentiment information in review text, and the impact of geographical locations. Wang et al. [35] proposed a multi-source information fusion recommendation approach that integrates social trust information, reviews, and other information into a unified model using Collaborative Filtering. However, feature-based multi-source information fusion often relies on domain experts with extensive experience and knowledge, requiring substantial time and effort for feature selection, extraction, and transformation.

Graph neural networks are capable of modeling connections between multiple sources of data, particularly in modeling item-to-item and user-to-user relationships, as well as handling explicit and implicit information. They enable effective fusion of multi-source data and enhance recommendation accuracy. Wang et al. [36] utilized a Heterogeneous Information Network to adaptively fuse rich content information, including metadata, tags, lyrics, and music text content, as well as contextual data to create a personalized music recommender system. Sun et al. [37] applied a knowledge graph to introduce various modal information as separate entities into the original knowledge graph and utilized aggregated embedding representations for recommendation. In complex recommendation scenarios, many research efforts have proposed using hypergraphs to model multi-source information [38], [39]. While ensuring information integrity, hypergraphs can effectively model high-order relationships, thus achieving better information fusion. The HEMR model [22] was proposed to model the association between user behavior information and music metadata through hypergraphs. It utilizes an improved random walk algorithm and Word2Vec to learn nodes embedding information, further mining user preferences, and had a significant improvement over the baseline.

Existing research highlights the advantages of hypergraph fusion for multi-source information integration and establishing user-item relationships in recommender systems. In general, recommendation performance based on hypergraphs depends on the construction of the hypergraph and hypergraph representation learning algorithms. This paper introduces a hypergraph structure based on composite hyperedges to integrate multi-source heterogeneous graphs, enabling in-depth exploration of high-order semantic correlations between data and facilitating improved accuracy in subsequent recommendation tasks.

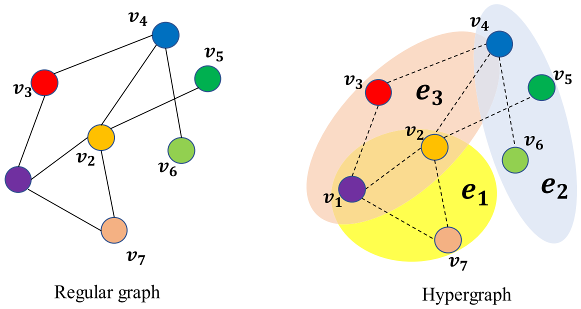


FIGURE 1. Illustrates examples of regular graphs and hypergraph structures.

B. HYPERGRAPH CONSTRUCTION METHODS

Regular graphs are typically used to describe a set of objects with binary relationships. The main distinction between hypergraphs and regular graphs lies in the number of nodes connected by each edge. In hypergraphs, each edge can connect multiple nodes, representing associations that are not limited to pairwise relationships, thus capturing more complex higher-order connections, as illustrated in Figure 1.

As a means to effectively explore and establish data relationships, hypergraphs are applied including image processing tasks and recommender systems. In general, the core idea of existing hypergraph construction methods is to mine connections from data to generate hyperedges. For example, Jin et al. [40] used a specific class feature buffer in computer vision tasks and selected features from the corresponding class-specific feature buffers based on k-nearest neighbors to further generate hypergraph feature hyperedges. On the other hand, Zhang et al. [41] predicted whether hyperedges exist within node groups based on the pseudo-distance between hypergraph node embeddings. Wang et al. [42] utilized a representation-based hypergraph generation method to determine relationships between vertices through feature reconstruction for hypergraph generation. However, methods based on feature distances require a certain level of accuracy in feature extraction representation for unstructured data. Moreover, hypergraph construction methods based on attributes [43] and network structures like social networks simplify the hypergraph construction process by utilizing user or item attributes and existing network structure relationships. Recently, Gao et al. [44] employed an adaptive hyperedge group fusion strategy on top of graph representation and feature distance-based hyperedge construction methods, to effectively fuse different hyperedge groups and construct hypergraphs. Unlike various hypergraph constructions mentioned above, multiple hyperedge groups can preserve more associated information. This multi-hyperedge group approach can mine hidden information, enhancing the accuracy of downstream tasks.

In recommendation tasks, hypergraphs are capable of modeling complex relationships and connection patterns, extending beyond simple user-item pairs. Xia et al. [45] aligned social network graphs with user-item bipartite graphs to discover semantic relationships and generate hypergraphs. In real-world scenarios with multiple data sources, as opposed to structured data, the existence of unstructured data such as audio, video, images, and text poses a challenge for existing methods to represent multi-source information in a unified manner. Some researchers have attempted to handle multi-source data using methods that involve multiple hypergraphs. For example, He et al. [46] created three hypergraphs to integrate multimodal data from user-item relationships and associated attributes, aiming to enhance recommendation accuracy by learning perceptual representations of users and item representations based on multiple hypergraphs. Kim et al. [47] developed a hypergraph attention network to balance information levels across different modalities of multi-source information, constructing a common semantic space using symbolic graphs from each modality. However, building multiple hypergraphs to address multi-source information fusion may lead to missing connections between nodes with multivariate relationships, thus impacting the performance of recommender systems.

In hypergraph-based recommendation approaches, existing research relies on social graphs and bipartite graph alignment to construct hypergraphs through a single method, such as distance-based methods, resulting in certain limitations related to missing multi-source information associations. Based on multi-source data fusion, we connect multiple heterogeneous graphs through composite hyperedges. This method not only facilitates the modeling of high-order associations but also aids in the extraction of both explicit and implicit user-item information.

C. HYPERGRAPH LEARNING

Hypergraph learning aims to capture the topological structure of a graph, relationships between nodes, and information related to graphs, subgraphs, and nodes by mapping hypergraph data into low-dimensional dense vectors. Traditional hypergraph learning often transforms hypergraphs into ordinary graphs through techniques like sampling expansion and uses graph processing methods such as Graph Convolutional Networks (GCN) [48] to learn graph embeddings. However, this approach often leads to the loss of nodes and structural information. Carletti et al. [49] introduced a hypergraph random walk algorithm driven by a Laplacian operator directly on the hypergraph model. They experimentally demonstrated that such non-expansion-based random walks preserve high-order structural information more effectively in hypergraphs. Gao et al. [20] designed tensor representations to flexibly represent dynamic hypergraph structures and introduced a method for effectively learning dynamic hypergraphs known as Tensor-Based Dynamic Hypergraph

Learning. In contrast to employing an adjacency matrix, optimizing tensor representations enables adjustments in both weights and the number and order of hyperedges.

Currently, hypergraph learning modules utilizing deep learning techniques have gained significant attention. HGNN [24] was proposed based on GCN and hypergraph Laplacians, which enables spectral convolution on hypergraphs for fine-grained feature extraction of nodes and hyperedges. More recently, Gao et al. [44] defined two hypergraph convolutions from spectral and spatial perspectives, respectively, to further improve the HGNN. Due to the flexible application of attention mechanisms, Bai et al. [50] combined hypergraph convolutional networks with an attention mechanism to recompute adjacency matrices for hypergraphs, enhancing representation learning, albeit limited to uniform hypergraphs. Yi and Park [51] attempted to combine RNN networks with hypergraph convolutional networks to learn temporal dependencies in sequence data with fewer parameters, and proved that the HGC-RNN accurately captures the data's spatial and time-based progression.

In the application of hypergraph learning to recommendation tasks, Karantaidis et al. [52] proposed an optimization method spanning multiple stages using hypergraphs. By the optimization of hypergraph ranking, hypergraph updates, and adaptive edge weight, accurate ranking vectors generated for image and label recommendation. Wang et al. [53] constructed a hypergraph model for users and evaluation items and introduced the constructed hypergraph by dynamic clustering method. This method clustered highly related users into the same interest communities to learn users' dynamic preferences. However, Ji et al. [54] created a hypergraph model for users and items separately called dual-channel hypergraph model (DHCF) and combined them with collaborative filtering principles to complete recommendation tasks. Similarly, Yuan et al. [55] went a step further by establishing hypergraph models for users and items, then improving hypergraph convolutional networks using contrastive learning and feature cross methods to achieve the final embeddings.

Hypergraph neural networks predominantly focus on information propagation between nodes and hyperedges, with less emphasis on interactions between nodes and hyperedges or the spatial density information interwoven between hyperedges. Additionally, there is a notable absence of attention towards combining hypergraph convolutional networks with GRU networks to explore issues related to the evolution of user preferences over time. Thus, this study introduces spatial density information and proposes an SD-HGAT model that combines GRU networks to delve deeper into hypergraph information and extract users' short-term preferences.

D. GRAPH CONTRASTIVE LEARNING

Graph contrastive learning has excellent performance in solving complex relationship modeling and data missing in recommender systems. Through contrastive learning, recommendation models can generate more discriminative representations of users and items, discover latent shared features, and utilize them for better data modeling and prediction. Contrastive learning has shown promising results in various fields, including graph representation learning [56], [57], [58] and recommender systems [59], [60]. Among them, multi-view contrastive learning allows for richer and more comprehensive information extraction from different views or data sources, making it suitable for handling diverse sources of heterogeneous information and facilitating personalized knowledge transfer tailored to individual user preferences.

Recent research in recommender systems employing graph-based contrastive learning aims to enhance information sharing across various hierarchical levels. For instance, Yu et al. [61] established a hypergraph model to mine user relationships and leveraged multi-channel hypergraph convolutional networks and self-supervision to enhance social recommendation. Moreover, Xia et al. [62] introduced Hypergraph Contrastive Collaborative Filtering (HCCF) that combined contrastive learning with recommender systems. HCCF utilizes an improved cross-view contrastive architecture to mine different levels of collaborative relationships. Zou et al. [63] designed global-level structure views, local-level collaboration views, and semantic views. They implemented contrastive learning across three different perspectives, encompassing both local and global levels, to capture a comprehensive range of graph information. Additionally, they introduced a project-project module to mine important semantic relationships that are often overlooked in previous research.

Presently, most approaches to constructing multi-view contrastive learning for hypergraphs are based on node perturbations, hyperedge perturbations, and subgraph generation [64]. Cai et al. [65] enriched data representations by extracting nodes and hyperedges from hypergraphs to generate different views. A hierarchical self-supervised model [66] was proposed, which further cross-matches different views to capture the contrast between information within and between various topics. They demonstrated that directly performing contrastive representation on multi-channel data makes the information highly homogeneous, causing a loss of uniqueness. Such a contrastive method has a negative impact and reduces recommendation performance.

In summary, to address the challenges of modeling complex relationships and sparse data in multi-variable relationships, this study designs a multi-view contrastive learning approach centered around a knowledge graph. Additionally, considering the uniqueness of features across multiple views, we cross match the knowledge graph with bipartite graphs, sequence graphs, and social graphs, and use the local properties of the four views and the global nature of multi-source heterogeneous hypergraphs to complete contrastive learning.

III. METHOD

The overall architecture of the proposed recommendation algorithm, named Multi-source Heterogeneous Hypergraph Contrastive Learning for Recommender System, is depicted

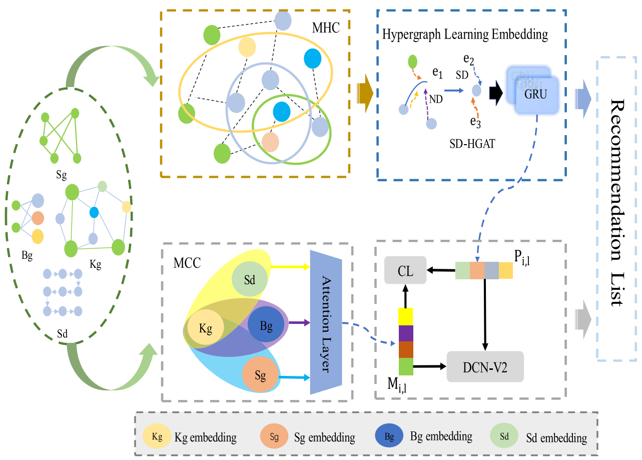


FIGURE 2. The overall framework of MHCLR.

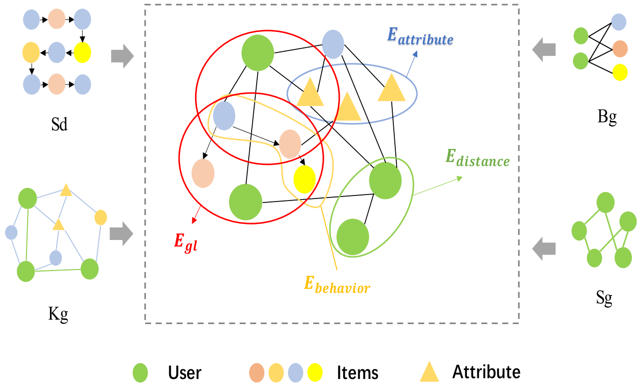


FIGURE 3. Construction of multisource heterogeneous hypergraph.

in Figure 2. The process is split into four steps: (1) The process of fusing composite hyperedges is utilized to generate multi-source heterogeneous hypergraphs, which are constructed upon the foundations of various graph types, including Bg, Sd, Sg, and Kg; (2) SD-HGAT and GRU networks are employed to extract user and item features, facilitating an in-depth exploration of user preferences; (3) Developing a multi-view cross-contrast learning module improves feature extraction precision, refining multivariate relationship modeling and enriching multi-perspective feature information; (4) A feature cross-network is utilized to compute high-order features for users and items. Similarity calculations between users and items are performed at different feature scales, resulting in the generation of a Top-k recommendation list.

A. HYPERGRAPH CONSTRUCTION

Integrating multiple information sources makes hypergraph models more complex, but it also improves the comprehensiveness of user profiling and the accuracy in revealing both explicit and implicit preferences. This paper proposes a method for generating attribute hyperedges on both the user and item sides, which is based on the alignment of user-item attributes Bg, Sg, Sd and Kg. We design behavior hyperedges with sequential information, derived from the analysis of user-item interaction data. The KNN algorithm is utilized to aggregate closely related features, which are then used to construct distance hyperedges. Prediction Hyperedges are employed to assess the associations between global and local nodes, forming additional hyperedges. We fuse these composite hyperedges to generate a comprehensive multi-source heterogeneous hypergraph.

Definition 1 Multi-Source Heterogeneous Hypergraph: The hypergraph is denoted as $\mathrm{G_m} = (\mathrm{V, E, W})$, where $\mathrm{V} = \{\mathrm{v_1, v_2, \cdots, v_m}\}$, $\mathrm{E} = \{\mathrm{e_1, e_2, \cdots, e_n}\}$, and $\mathrm{W} = \{\mathrm{w_1, w_2, \cdots, w_n}\}$ represent the sets of nodes, hyperedges, and weights, respectively. The hypergraph G_m can be represented by an $|\mathrm{V}| \times |\mathrm{E}|$ incidence matrix H, defined as:

$$\mathrm{H}(\mathrm{v, e}) = \begin{cases} 1, & \text{if } \mathrm{v} \in \mathrm{e} \\ 0, & \text{if } \mathrm{v} \notin \mathrm{e} \end{cases} \tag{1}$$

In recommendation tasks, we consider the user set $\mathrm{U} = \{\mathrm{u_1, u_2, \cdots, u_m}\}$ and the item set $\mathrm{I} = \{\mathrm{i_1, i_2, \cdots, i_n}\}$, where U is the set of individual users, and I is the set of individual items. In fact, hypergraphs also encompass node types such as user and various attributes related to items. $\mathrm{V} = \mathrm{U} \cup \mathrm{I}$ is the set of all nodes in the system. Based on multi-source data, we utilize a composite hyperedge construction method to generate the hypergraph, as illustrated in Figure 3. The specific process of hypergraph construction is as follows:

Definition 2 Composite Hyperedge E:

$$\mathrm{E} = \left\{\mathrm{E_{attribute}, E_{behavior}, E_{distance}, E_{gl}}\right\} \tag{2}$$

(1) Attribute Hyperedge Set $E_{attribute}$: Based on users and items attribute information, such as time information, geographic location, etc. relevant nodes are connected through attribute hyperedges denoted by "a". The vertices in hyperedge "a" share the same attribute type, and V_{att} (a) represents the set of nodes included in hyperedge "a". A represents the collection of all attribute hyperedges.

$$\mathrm{E_{attribute}} = \{\mathrm{V_{att}(a)} \mid \mathrm{a} \in \mathrm{A}\} \tag{3}$$

(2) Behavioral Hyperedge Set $E_{behavior}$: Based on the user-item historical interaction records, such as a collection of all interactions between a user and a specific item. Behavioral hyperedges denoted by "b" are defined to connect user and item nodes. V_{be} (b) represents the set of nodes that includes user and item information, and B is the collection of all behavioral information.

$$\mathrm{E_{behavior}} = \{\mathrm{V_{be}(b)} \mid \mathrm{b} \in \mathrm{B}\} \tag{4}$$

(3) Distance Hyperedge Set $E_{distance}$: In the context of heterogeneous networks with multiple sources of information, for a given node v, we can define a distance hyperedge e in the feature space, which connects node v to its k nearest neighbor nodes in the feature space with a distance metric. We can define a distance hyperedge denoted by "d" that connects the node to its k nearest neighboring nodes. V_{dis} (d) represents

feature nodes information, and D is the aggregation of all distance hyperedges.

$$\mathrm{E_{distance}} = \{\mathrm{V_{dis}}(\mathrm{d})\,|\mathrm{d} \in \mathrm{D}\} \tag{5}$$

Definition 3 Prediction Hyperedge: This is achieved through a designed computational method aimed at obtaining pseudo-distances between node local and global embeddings, thereby predicting the probability of the existence of hyperedges among groups of nodes [41]. Unlike dynamic-static node embeddings, we focus more on updating the local information of nodes. Nodes within the first-order neighborhood of the target node have strong correlations, and pseudo-distances between global node embeddings can better reflect the similarity within the node group.

The computation process of SAGGL: (1) Local Nodes: Given the input m nodes group representation $\{\mathrm{v_1}, \mathrm{v_2}, \cdots, \mathrm{v_m}\}$, the GNN network is used to update the information of the target node v_i based on its first-order neighborhood. This results in the local embedding l_i. (2) Global Nodes: Utilizing a self-attention mechanism layer to calculate the connections between the target node v_i and other nodes in the input node group representation $\{\mathrm{v_1}, \mathrm{v_2}, \cdots, \mathrm{v_m}\}$, providing a representation of global information denoted as g_i. (3) Using the power Hadamard product of the difference between the local embedding and global embedding vectors for each node, followed by a neural network with ReLU as the activation function, to obtain a probability score p_i. The average of all probability scores yields P, representing the probability of the existence of hyperedges within that group of nodes. o_i can be seen as the squared weighted pseudo-Euclidean distance between the local embedding l_i and global embedding g_i, with σ denoting the ReLU activation function.

$$\mathrm{o_i} = \mathrm{W_o^T}\left((\mathrm{g_i} - \mathrm{l_i})^{\mathrm{o}2}\right) + \mathrm{b} \tag{6}$$

$$\mathrm{P} = \frac{1}{\mathrm{m}}\sum\nolimits_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{p_i} = \frac{1}{\mathrm{m}}\sum\nolimits_{\mathrm{i}=1}^{\mathrm{m}} \sigma(\mathrm{o_i}) \tag{7}$$

(4) Prediction Hyperedge Group E_{gl}: Using the SAGGL algorithm to predict potential hyperedges denoted as s, enhancing the information within the multi-source heterogeneous hypergraph. S represents the collection of all predicted hyperedges, and V_{gl} comprises the nodes contained within such hyperedges.

$$\mathrm{E_{gl}} = \left\{\mathrm{V_{gl}}(\mathrm{s})\,|\mathrm{s} \in \mathrm{S}\right\} \tag{8}$$

Considering the differences in information among various types of hyperedge groups and within the same type of hyperedge group, simple equal-weight hyperedge fusion cannot fully leverage the high-order correlations among multiple sources of information. Therefore, in this paper, we utilize an adaptive hyperedge weight fusion method [44], defined as: $\mathrm{w_k} = \mathrm{copy}(\mathrm{sigmoid}(\mathrm{w_k}), \mathrm{M_k})$, the weight parameter for each type of hyperedge group, $\mathrm{W} = \mathrm{diag}\left\{\mathrm{w_{att}^{k_a}}, \mathrm{w_{be}^{k_b}}, \mathrm{w_{dis}^{k_d}}, \mathrm{w_{gl}^{k_{gl}}}\right\} \in R^{M\times M}$ is a diagonal matrix representing the weight matrix of the hypergraph, where each w_e^i represents the weight of the corresponding hyperedge. $w_{att}^{k_a}, w_{be}^{k_b}, w_{dis}^{k_d}, w_{gl}^{k_{gl}}$ represent the weight matrices for the attribute hyperedge group, behavior hyperedge group, distance hyperedge group, and prediction hyperedge group, respectively. Sigmoid is an element-wise normalization function. $\mathrm{H} = \{\mathrm{H_1}||\mathrm{H_2}||\cdots||\mathrm{H_m}\}$ is the adjacency matrix of the multi-source heterogeneous hypergraph formed by connecting the composite hyperedge groups with "||". To emphasize user preferences for certain items, we extract the interaction frequency matrix for each user-item pair and use it as the initial weight matrix for the behavior hyperedge to enhance the model performance.

B. HYPERGRAPH LEARNING

In order to better handle multi-source heterogeneous hypergraphs and obtain accurate features, we propose SD-HGAT, which combines spatial density information with hypergraph attention networks as shown in Figure 4. To simplify the efficiency of SD-HGAT operations, we design a single-layer hypergraph convolution to obtain refined feature representations h_j^{l-1} for complex multi-source heterogeneous hypergraph nodes. We then fuse node density information with spatial density information to obtain hyperedge density information, which is used to obtain the node embedding features h_j^l for the next layer. Through the SD-HGAT network layers, we obtain user/item feature embeddings as input to the GRU network, where we learn user preference information to enhance recommendation accuracy.

Hypergraph learning primarily deals with the effective propagation of information among adjacent nodes and structural information. HGCN focuses on aggregating node information into hyperedges and aggregating hyperedge information into nodes. The hypergraph convolution layer is defined by the formula $\mathrm{f}(\mathrm{X}, \mathrm{W}, \theta)$:

$$\mathrm{X^{(l+1)}} = \sigma\left(\mathrm{D_v^{-1/2}HWD_e^{-1}H^TD_v^{-1/2}X^{(l)}}\theta\right) \tag{9}$$

The symbol σ represents a non-linear activation function. Where D_e and D_v represent the diagonal matrices for hyperedge degrees and vertex degrees, respectively. θ is a trainable weight matrix. Here, $X^{(l)}$ represents the signal at the l-th layer of the hypergraph, with $X^{(0)}$ denoting the initial nodes input.

Definition 4 Node-Level Density Attention: Node-level density attention associates density information for each vertex to obtain representations of hyperedges. First, we define the density of each node, which can be defined as the sum of similarities between adjacent nodes whose similarity to the target node exceeds a predefined threshold. The density of node x_i can be expressed as:

$$\rho_{\mathrm{x_i}} = \sum\nolimits_{\mathrm{x_k}\in \mathrm{N}(\mathrm{x_i})} \begin{cases} \mathrm{sim}(\mathrm{x_i}, \mathrm{x_k}), & \text{if } \mathrm{sim}(\mathrm{x_i}, \mathrm{x_k}) > \delta \\ 0, & \text{if } \mathrm{sim}(\mathrm{x_i}, \mathrm{x_k}) \le \delta \end{cases} \tag{10}$$

where x_i represents the feature vector of v_i and $N(x_i)$ represents the neighborhood of node x_i. δ is a predefined threshold.

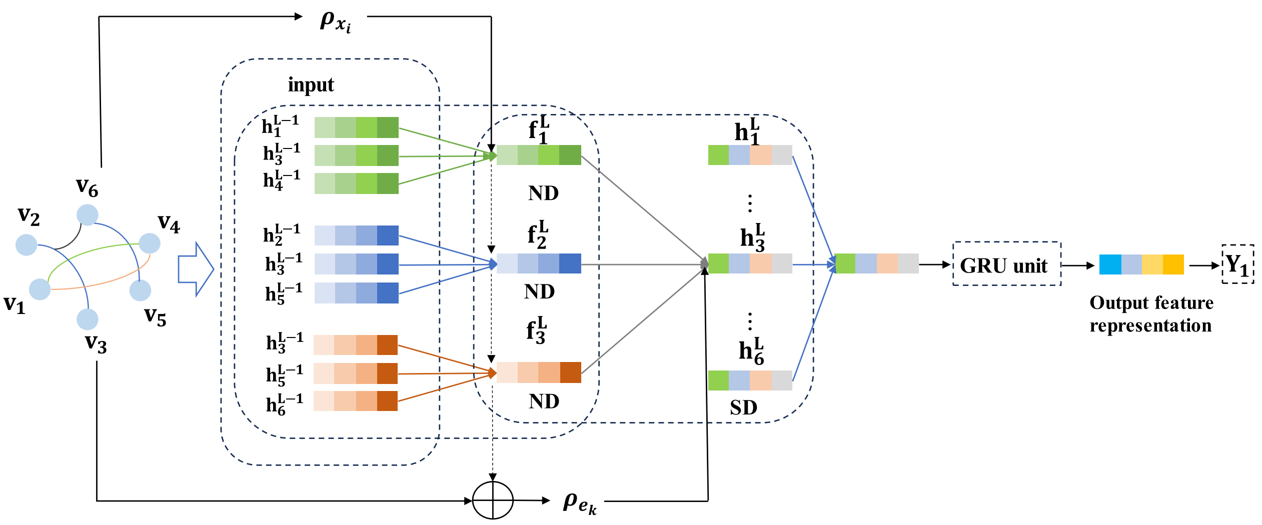


FIGURE 4. Hypergraph learning module.

sim is a similarity measure function, typically cosine similarity can be used. We utilize Hyper-GAT to obtain the hyperedge representation of node v_i. Since the contribution of nodes to the hyperedge e_j varies, an attention mechanism is used to highlight those nodes that are important for the hyperedge. These nodes are aggregated to obtain the representation of the hyperedge:

$$f_j^l = \sigma\left(\sum\nolimits_{v_k\in e_j}\left(\alpha_{jk}+\rho_{x_i}\right)W_1 h_k^{l-1}\right) \tag{11}$$

where σ is a nonlinear function, such as ReLU, and W_1 is a trainable weight matrix. α_{jk} represents the attention coefficient for node v_k in hyperedge e_j, which can be calculated using the following formula:

$$\alpha_{jk} = \frac{\exp\left(a_1^T u_k\right)}{\sum_{v_p\in e_j}\exp\left(a_1^T u_p\right)} \tag{12}$$

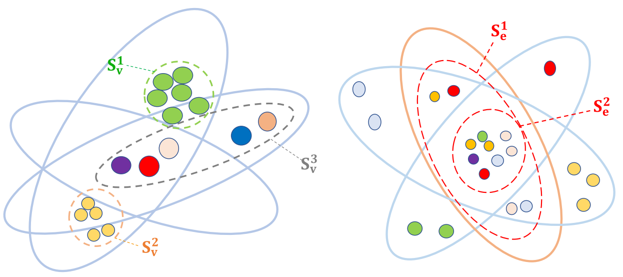
$$u_k = \mathrm{LeakyReLU}\left(W_1 h_k^{l-1}\right) \tag{13}$$

Among them, a_1^T is the weight vector. u_p refers to the correlation degree of node p on the hyperedge e_j.

Definition 5 Hyperedge-Level Density Attention: After obtaining the aggregation results for all hyperedges, we apply the attention mechanism again to highlight meaningful hyperedges and learn the representation of the next layer of nodes x_i. First, we need to define a density for each hyperedge. The difference lies in that the density of each hyperedge is defined as the sum of the densities of all nodes connected by that hyperedge, weighted by certain values, plus the ratio of nodes shared between hyperedges in space to all nodes of the hyperedge. This can be formulated as:

$$\rho_{e_k} = a_1\frac{\left|V(e_i)\cap V(e_j)\right|}{\left|V(e_i)\cup V(e_j)\right|} + a_2\sum\nolimits_{x_i\in N(e_i)}\rho_{e_i} \tag{14}$$

where a_1 and a_2 are the weight coefficients for spatial density and hyperedge density, respectively. $V(e_i)$ represents all nodes on the hyperedge e_i. For all hyperedge representations $\{f_j^l|e_j\in\varepsilon_i\}$, we once again apply the hyperedge-level attention mechanism to emphasize the hyperedges containing information relevant to learning the next-layer representation of node v_i. This process can be expressed as:



(a) Node-level density

(b) Hypergraph-level density

FIGURE 5. Diagrams illustrating node and hyperedge density. (a) Node-level density, where S_v^1 and S_v^2 represent regions of node density under the threshold δ, and S_v^3 represents a sparse region. (b) Hyperedge-level density diagram, whereS_e^1 represents the intersection density region between two types of hyperedges, and S_e^2 represents the sum of node densities.

$$h_j^l = \sigma\left(\sum\nolimits_{e_j\in\varepsilon_i}\left(\beta_{ij}+\rho_{e_k}\right)W_2 f_j^l\right) \tag{15}$$

In which, h_j^l represents the output representation of node x_i, W_2 is the weight matrix. β_{ij} denotes the attention coefficient of hyperedge e_j on node x_i, and it can be calculated using the following formula:

$$\beta_{ij} = \frac{\exp\left(a_2^T x_j\right)}{\sum_{e_p\in\varepsilon_j}\exp\left(a_2^T x_p\right)} \tag{16}$$

$$x_j = \mathrm{LeakyReLU}\left(\left[W_2 f_j^l || W_1 h_k^{l-1}\right]\right) \tag{17}$$

Among them, a_2^T is another weight vector used to measure the importance of hyperedges.

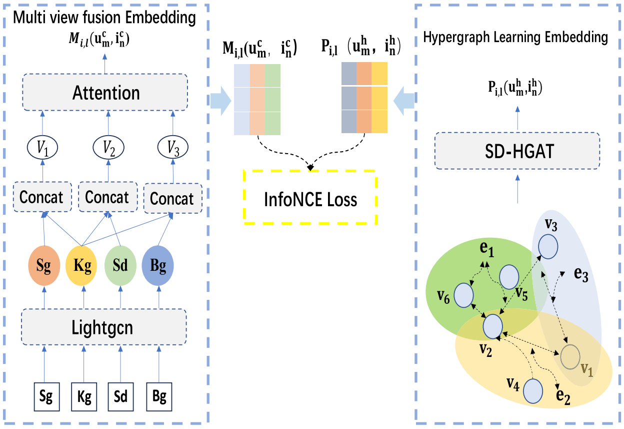


FIGURE 6. Contrastive learning module.

The SD-HGAT proposed in this paper integrates node density and hyperedge density into node-level attention and hyperedge-level attention networks, highlighting different granularity of key information in the node representation learning process, as shown in Figure 5. In order to comprehensively capture user preferences, GRU is utilized to extract the latent interest states of users from their historical behavior sequences. Through SD-HGAT and GRU, we obtain refined feature vector representations for users, denoted as u_m^h, representing the m-th user vector after hypergraph-based learning, and i_n^h representing the n-th item vector after hypergraph-based learning.

C. MULTIPLE CROSS VIEW CONTRASTIVE LEARNING

In response to the challenges of data sparsity and the complexity of modeling complex relationships in real-world recommendation scenarios with hypergraph structures, this paper proposes a multi-view cross-fusion contrastive learning module to improve recommendation accuracy and diversity through multi-level features. The contrastive learning module, as shown in Figure 6, consists of two specific modules: multi-view cross-fusion and contrastive learning modules.

Definition 6 Multi-View Cross-Fusion: Multi view contrastive learning is crucial for recommendation performance. This article adopts a cross fusion approach, exploring the inherent connections between different views to avoid the problem of information redundancy in multi-source heterogeneous graphs that may not be effectively solved by direct fusion, ensuring that the model can more accurately distinguish subtle differences between different views. This type of cross view contrastive learning can avoid homogenization of information between multiple views, enhancing the model's comprehensive understanding and prediction of user preferences. By encouraging the model to learn more independent and discriminative feature representations, the accuracy of recommendation results has been further improved. In the construction of the multi-source heterogeneous hypergraph, we utilize the generated knowledge graph, bipartite graph, social graph, and sequential graph as multiple views of the hypergraph data. We use LightGCN [67] to capture representations of the four views: knowledge graph, bipartite graph, social graph, and sequential graph. We fuse the representations of the four views using an attention mechanism and then perform contrastive learning with the final embeddings of the multi-source heterogeneous hypergraph. In the graph encoding module, given these multiple views, we utilize graph convolutional neural networks to encode the multi-views. The graph convolutional neural network is defined as follows:

$$\mathrm{V_{fusion}} = \alpha_1 \mathrm{V_1} + \alpha_2 \mathrm{V_2} + \alpha_3 \mathrm{V_3} \tag{18}$$

$$\alpha_\mathrm{i} = \frac{\mathrm{a_i}}{\sum_{\mathrm{j}=1}^{3} \exp(\mathrm{a_j})} \tag{19}$$

Among them, a_i is the calculated V_i of the view. The attention score is obtained through conventional attention calculation. V_{fusion} represents the fused nodes representation, $\mathrm{V_i}$ represents the fusion of Kg and Bg, Sd, or Sg, using feature vector concatenation.

$$\mathrm{L_s^{(u)}} = \sum\nolimits_{\mathrm{i}=0}^{\mathrm{I}} \sum\nolimits_{\mathrm{l}=0}^{\mathrm{L}} -\log \frac{\exp(\mathrm{s}(\mathrm{M_{i,l}^{(u)}}, \frac{\mathrm{P_{i,l}^{(u)}}}{\tau})}{\sum_{\mathrm{i'}=0}^{\mathrm{I}} \exp(\mathrm{s}(\mathrm{M_{i,l}^{(u)}}, \frac{\mathrm{P_{i,l}^{(u)}}}{\tau})} \tag{20}$$

where $s(\cdot)$ represents the cosine similarity function, and τ is a tunable hyperparameter that adjusts the softmax scaling.

We perform contrastive learning between the multi-view cross-fusion embedding $M_{i,l}$ and the hypergraph-guided representation $P_{i,l}$. This allows local and global dependency views to mutually supervise each other, enhancing the representations of users and items. Node feature information from multiple perspectives can be obtained, thereby improving their accuracy in downstream tasks.

D. RECOMMENDATION GENERATION

To capture more implicit interaction information between users and items, we design a feature cross layer to perform feature cross between the user vectors u_m^h and item i_n^h from the hypergraph learning module and the nodes representations u_m^c and i_n^c generated by the contrastive learning module. We introduce $\{\mathrm{u_m^h}, \mathrm{i_n^h}, \mathrm{u_m^c}, \mathrm{i_n^c}\}$ into the DCN V2 [68] model to facilitate feature co-occurrence and enhance the model non-linearity. This enables the recommendation task module to leverage the data features learned in the contrastive learning module and enriches the semantic features of the node representation vectors generated in this module, thus improving the model's recommendation performance and diversity of results. Considering that suboptimal results from the feature cross layer may negatively affect the semantic features of the nodes representation vectors generated by the recommendation task module, we add standard residual connections in the network to preserve the node feature representations learned from both contrastive learning and hypergraph learning. This enhances the generalization capability of the node representation vectors. The formula for the feature cross network is as follows:

$$\mathrm{x_{l+1}} = \mathrm{x_k} \odot \{\mathrm{W_l}\,(\mathrm{x_k} + \mathrm{x_m}) + \mathrm{b_l}\} \tag{21}$$

$$\mathrm{x_{out}} = \mathrm{x_{l+1}} + \mathrm{W_{Res}} \mathrm{x_0} \tag{22}$$

where x_0 represents the original features, x_k and x_m represent the joint input from the l-th layer of hypergraph learning and contrastive learning, x_{l+1} indicates the output of the l+1-th layer, W_l is the weight matrix to be learned, b_l is the bias vector, and W_{Res} is a projection matrix for dimension mismatch situations.

The feature cross layer results in high-order user representation u_m^{fc} and item representation i_n^{fc}. Finally, we calculate the similarity between them using the cosine function to predict the probability of a user choosing a particular item. To effectively utilize high-order feature combinations and hypergraph learning nodes feature information, we design two rounds of similarity calculations: the first round y_1 is based on the node representation from hypergraph learning, and the second round y_2 is based on the high-order feature representations obtained from feature cross. The two similarity calculation results are then softmax-normalized to obtain the final prediction. The Top-k items based on the predicted probabilities form the recommendation list.

$$\mathrm{y}_1 = \cos\left(\mathrm{u}_{\mathrm{m}}^{\mathrm{h}}, \mathrm{i}_{\mathrm{n}}^{\mathrm{h}}\right) \tag{23}$$

$$\mathrm{y}_2 = \cos\left(\mathrm{u}_{\mathrm{m}}^{\mathrm{fc}}, \mathrm{i}_{\mathrm{n}}^{\mathrm{fc}}\right) \tag{24}$$

Final probability prediction:

$$\mathrm{Y} = \mathrm{softmax}\left(\mathrm{z}_1\mathrm{y}_1 + \mathrm{z}_2\mathrm{y}_2\right) \tag{25}$$

Among them, z_1, z_2 are the adaptive weight for two recommended predictions.

E. COMPLEXITY ANALYSIS

The main computational cost of the MMCLR model is composed of multi-source heterogeneous hypergraph construction, hypergraph density attention network, contrastive learning, and feature crossover. In hypergraph construction, nodes and hyperedges are the main influencing factors, with a complexity of $O(N||C)$, where N is the number of nodes in the hypergraph and C is the number of hyperedges in the hypergraph. After passing through an L-layer hypergraph density attention network, the model consumption is less than $O(N^2| |\mathrm{K}| \,\mathrm{dL})$, where N^2 is the number of common nodes in density attention calculation and K is the number of attention heads, where K has a lower number of heads. Contrastive learning is the comparison between multi view fusion and hypergraph, with a complexity of $O = (L \times (D + M) \times d)$. Among them, $D + M$ is the sum of positive and negative samples constructed. The complexity of the feature crossover layer is $O(\sum_{Ld} d_{l+1}d_l)$, d_l is the size of the l-th feature intersection layer. So, the overall complexity of our model is $O(N||P+N^2| |\mathrm{K}| \,\mathrm{dL_h}+L_c \times (D+M) \times d)+ O(\sum_{Ld} d_{l+1}d_l)$.

IV. EXPERIMENT

In this section, we will present the experiments conducted on a demonstration dataset to evaluate the performance metrics of the proposed method. We will also analyze the obtained results and compare them with other state-of-the-art models in the field of recommender systems.

TABLE 1. Dataset statistics.

Datasets	Users	Items	Interactions	Density
Last-FM	1892	17632	92834	0.0028
Douban	2848	39586	35770	0.0079
Yelp	29601	24734	1517326	0.0011

A. EXPERIMENTAL SETUP

1) DATASETS

We utilized three publicly available datasets related to recommender systems, namely Last-FM, Douban, and Yelp, to evaluate the MHCLR model. The dataset statistics are summarized in Table 1. (1) Last-FM Dataset: This dataset includes information about users' social networks, tags, and their frequently listened-to music artists; (2) Douban Dataset: The Douban dataset, which has been widely used for various recommendation tasks, contains 2,848 users and 39,586 movies; (3) Yelp Dataset [69]: The Yelp dataset comprises business-related information, including businesses, user reviews of businesses, and users' social networks.

2) CONSTRUCTION OF HYPERGRAPH SOURCE DATA

(1) In the Last-FM dataset, three types of structures—social graphs, bipartite graphs, and knowledge graphs—were utilized. They were combined to construct a hypergraph using adaptive fusion through distance hyperedges, attribute hyperedges, and prediction hyperedges.

(2) In the Douban dataset, three types of structures—social graphs, bipartite graphs, and knowledge graphs—were employed. They were integrated to construct a hypergraph through adaptive fusion using distance hyperedges, attribute hyperedges, behavior hyperedges, and prediction hyperedges.

(3) In the Yelp dataset, four types of structures—social graphs, sequence graphs, bipartite graphs, and knowledge graphs—were used. These structures were combined to construct a hypergraph through adaptive fusion with the aid of distance hyperedges, attribute hyperedges, and prediction hyperedges.

3) BASELINE ALGORITHMS

To demonstrate the effectiveness of MHCLR, we compared it with four recommendation system methods: Collaborative Filtering (BPR), Graph Neural Network-based methods (LightGCN, GraphRec), Hypergraph Neural Network-based method (DHCF), and Contrastive Learning-based methods (MHCN, SGL). These are described as follows:

- BPR [70]: A commonly used recommendation algorithm that leverages implicit user feedback to rank items by maximizing the posterior probability obtained through Bayesian analysis of the problem.
- LightGCN [67]: Simplifies the framework by removing nonlinear projections and embedding transformations during message passing.

TABLE 2. Overview of baseline methods.

Methods	Data				Model			
	Interaction	Sequence	Social	KG	CF	Graph	Hypergraph	CL
BPR [70]	√	×	×	×	√	×	×	×
LightGCN [67]	√	×	×	×	√	√	×	×
GraphRec [71]	√	×	√	×	×	√	×	×
SGL [72]	√	×	×	×	×	√	×	√
MHCN [61]	√	×	√	×	×	×	√	√
DHCF [54]	√	×	×	×	√	×	√	×
MHCLR	√	√	√	√	×	√	√	√

TABLE 3. Performance comparison (%).

Data	Last-FM			Douban			Yelp		
Metrics	P@10	R@10	N@10	P@10	R@10	N@10	P@10	R@10	N@10
BPR [70]	15.606	15.821	18.953	15.673	5.160	17.476	2.002	5.173	3.840
GraphRec [71]	17.385	18.020	21.173	17.021	5.916	19.051	2.323	6.075	4.653
LightGCN [67]	19.205	19.480	23.392	17.780	6.247	19.881	2.586	6.525	4.988
SGL [72]	19.570	19.770	23.504	19.430	5.350	22.120	2.670	7.050	5.660
MHCN [61]	19.625	19.945	23.834	18.283	6.556	20.694	2.751	6.862	5.356
DHCF [54]	16.877	17.131	20.744	16.871	5.755	18.655	2.298	5.986	4.700
MHCLR	**19.820**	**20.376**	**24.132**	**19.613**	**6.252**	**21.142**	**2.728**	**7.102**	**5.702**

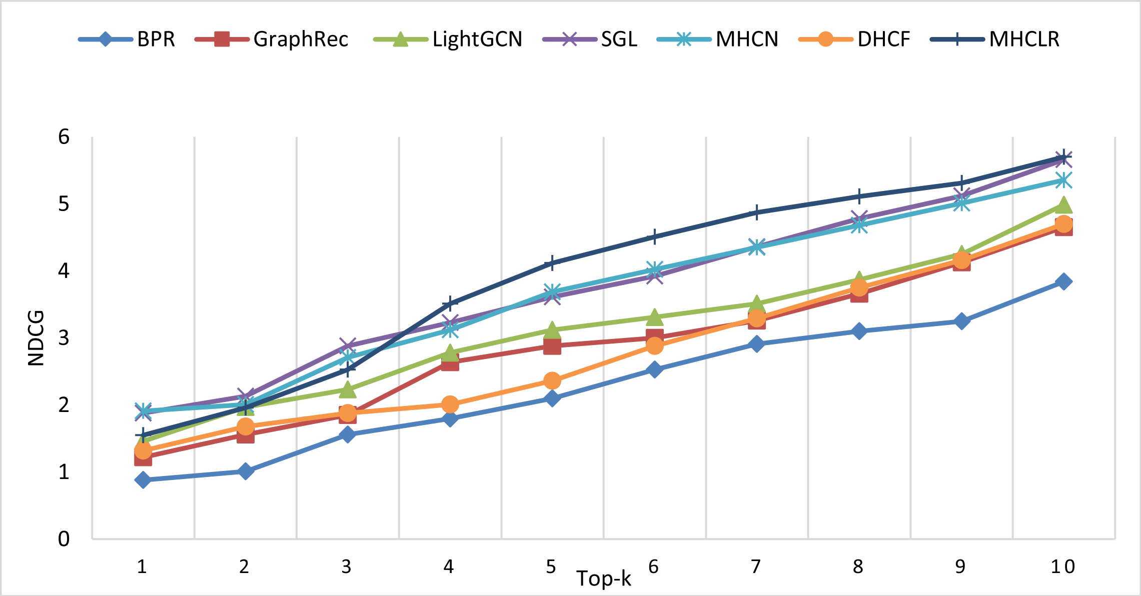


FIGURE 7. Top-k recommendation on Yelp dataset NDCG@k performance comparison.

- GraphRec [71]: A GNN-based social recommendation model that simultaneously models user-item and user-user interactions to capture user-item relationships and user opinions on items.
- SGL [72]: Augments LightGCN by generating multiple views through data augmentation using random walks and feature dropout, enhancing it with self-supervised contrastive learning.
- MHCN [61]: Maximizes the mutual information between node embeddings and global readout representations to regularize representation learning in interaction graphs.
- DHCF [55]: A novel hypergraph-based recommendation approach that models high-order relevance information using hypergraphs.

For fair comparison, we referred to the best parameter settings reported in the original papers of the baseline methods

TABLE 4. Contrastive learning performance on yelp.

Methods	P@10	R@10	N@10
MHR	1.552	4.232	2.880
MHR+NF	2.210	5.866	4.950
MHCLR	**2.728**	**7.102**	**5.702**

and then fine-tuned all hyperparameters of the baseline methods through grid search to ensure their optimal performance. The general settings for the models were empirical, with embedding dimensions set to 64 and a regularization coefficient of 0.001. Adam optimization was used for model training. The value of δ is set to 0.3. Detailed overviews of the baseline methods are provided in Table 2.

4) PERFORMANCE METRICS

The primary focus of this study is to provide Top-k recommendations to users. Therefore, commonly used recommendation system evaluation metrics, including Precision@K, Recall@K, and NDCG@K were employed. Precision@K measures the proportion of the Top-k recommended items that are actually liked by each user. It is calculated using the following formula:

$$\mathrm{Precision@K} = \frac{\sum_{\mathrm{u}\in\mathrm{U}} |\mathrm{R(u)} \cap \mathrm{T(u)}|}{\sum_{\mathrm{u}\in\mathrm{U}} |\mathrm{R(u)}|} \tag{26}$$

where R(u) represents the recommended list for user u based on their behavior in the training dataset, and T(u) corresponds to the list of user interactions in the test dataset.

The Recall describes the proportion of items in the recommended list that the user has previously interacted with, relative to the total number of items in the test dataset. The formula for Recall rate is as follows:

$$\mathrm{Recall@K} = \frac{\sum_{\mathrm{u}\in\mathrm{U}} |\mathrm{R(u)} \cap \mathrm{T(u)}|}{\sum_{\mathrm{u}\in\mathrm{U}} |\mathrm{T(u)}|} \tag{27}$$

NDCG@K is an evaluation indicator that considers the return order, with a value range of [0 1]. The formula is as follows:

$$\mathrm{NDCG@K} = \frac{\mathrm{DCG@K}}{\mathrm{IDCG@K}} \tag{28}$$

DCG is a measure of the ranking quality of recommendation results, while IDCG is the maximum cumulative gain that can be achieved by ranking from high to low according to correlation in an ideal state.

B. EXPERIMENTAL PERFORMANCE

In this section, we validate whether MHCLR outperforms existing recommendation baselines. The performance comparison of all methods on the Last-FM, Douban, and Yelp datasets is presented in Table 3.

Based on the results, we can draw the following conclusions:

(1) MHCLR demonstrates excellent performance in general recommendation tasks. Compared to LightGCN and DHCF, the MHCLR algorithm takes into account hypergraph density information. Additionally, contrastive learning significantly enhances MHCLR, alleviating dataset sparsity.

(2) In general recommendation tasks, GNN-based methods like GraphRec and LightGCN outperform BPR, especially with a noticeable improvement in NDCG@10. SGL, utilizing graph self-supervised learning, achieves better performance than LightGCN. Hypergraph-based recommendation method MHCN surpasses most GNN-based recommendation methods (GraphRec, LightGCN).

C. ABLATION EXPERIMENTS

In this section, we conduct ablation studies to investigate the interactions between different components of MHCLR and verify whether each component contributes positively to the final recommendation performance.

(1) Hypergraph Spatial Density Attention Network Investigation

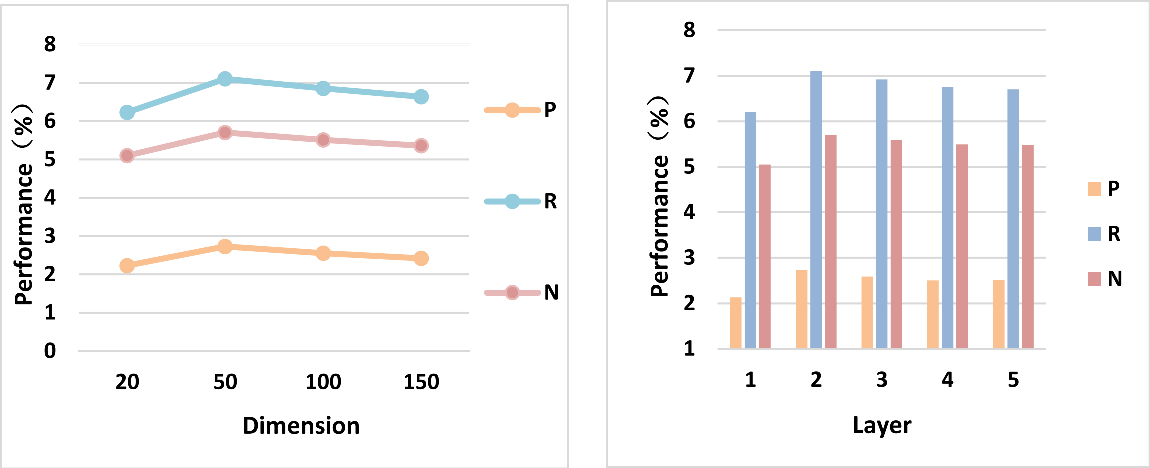
This subsection aims to delve into the performance differences between SD-HGAT, Hyper-GAT, and HGNN on hypergraph data by conducting ablation experiments on the hypergraph structure. To validate the effectiveness of spatial density information, we removed the contrastive learning module and solely utilize the original HGNN and Hyper-GAT, comparing their performance differences intuitively, as shown in Figure 9.

(2) Multi-View Contrastive Learning Performance Experiments

To assess the impact of the contrastive learning mechanism on recommendation performance, we attempt to remove the multi-view contrastive learning phase and evaluate its effect on performance through Multi-Source Heterogeneous Hypergraph Recommendation (MHR). We chose the Yelp dataset, which is closer to practical applications and involves multiple aspects of information, and it provides richer data on the interaction between users and items. Additionally, we test the influence of directly fusing multi-views and hypergraphs for contrastive learning (MHR+NF) to validate the importance of cross-view contrastive learning.

D. EXPERIMENTAL ANALYSIS AND DISCUSSION

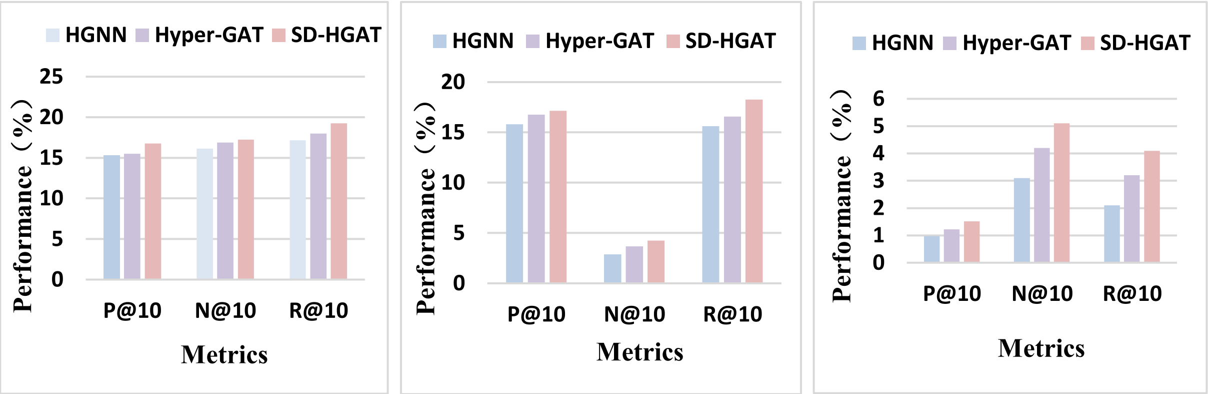
In this paper, we proposed the MHCLR model to provide accurate and personalized recommendation lists for recommender systems. We conducted experiments on the Last-FM, Douban, and Yelp datasets to demonstrate the effectiveness of the model. The results in Table 3 show that our MHCLR algorithm outperforms baseline models across all evaluation metrics. Specifically, on the Last-FM dataset, our algorithm improved P@10, R@10, and N@10 by 0.615%, 0.896%, and 0.74%, respectively, compared to the popular recommendation model LightGCN. It also outperforms the DHCF model based on hypergraph recommendation with improvements of 2.943%, 3.245%, and 3.388% in the mentioned metrics. On the Douban dataset, our algorithm achieved



(a) Embedding dimension

(b) Embedding Layer

FIGURE 8. Parameter sensitivity impact.



(a) Last-FM

(b) Douban

(c) Yelp

FIGURE 9. Performance comparison of HGNN, Hyper-GAT, and SD-HGAT.

improvements of 1.833%, 0.005%, and 1.261% in P@10, R@10, and N@10, respectively, compared to LightGCN, and outperformed DHCF with improvements of 2.742%, 0.497%, and 2.487%. On the Yelp dataset, our algorithm shows improvements of 0.142%, 0.577%, and 0.714% in P@10, R@10, and N@10, respectively, compared to LightGCN, and outperformed DHCF with improvements of 0.43%, 1.116%, and 1.002%. MHCLR algorithm outperforms LightGCN and SGL in recommendation performance, indicating that mining deeper user-item associations in the context of hypergraphs and contrastive learning can enhance the recognition of user preferences.

From this, we can conclude the following:

(1) As shown in Figure 7, when K is less than 5, the proposed MHCLR algorithm has slightly lower P, N, and R values than the optimal algorithm, which may be due to the impact of information loss. Compared to traditional graph structures, hypergraphs are more complex. When the k-value of the recommendation list is small, the number of nodes in the hypergraph may be fewer, which makes it difficult for hypergraph density attention networks to fully utilize the hypergraph structure. When generating recommendations, only a few candidate items are considered, which may lead to some potentially valuable recommendations being ignored and lacking sufficient recommendation coverage. As K increases, the performance of the MHCLR algorithm surpasses other algorithms, and the Precision and Recall gradually reach their peaks, achieving optimal algorithm performance. The MHCLR algorithm contributes to mining high-order relationships between users and items, enabling more accurate semantic descriptions of users and items. It delves deeper and comprehensively into user and item features, ultimately predicting user preferences more effectively.

(2) In this study, we constructed multiple SD-HGAT layers to simulate high-order information propagation among users in a hypergraph with high-order connections. This can be viewed as high-order information diffusion. We stacked

hypergraph attention network layers from 1 layer to 5 layers. As shown in Figure 8(b), the performance of MHCLR is optimal when the number of SD-HGAT network layers is 2. With an increase in the number of SD-HGAT layers, the performance of MHCLR decreases. This may be attributed to the excessive number of layers, which increases model complexity, leading to longer training times and requiring more computational resources. The 2-layer hypergraph attention network is relatively shallow, contributing to reducing the risk of overfitting and improving model generalization. Based on this analysis, the model achieves the best recommendation performance when the SD-HGAT network has 2 layers. Furthermore, we analyzed the embedding dimension d of the model. We set d to vary among {20, 50, 100, 150}. Figure 8(a) displays the performance metrics P@10, R@10, and N@10 on the Yelp dataset for different values of dimension. In general, as the embedding dimension increases, the model's performance initially rises and then declines. Significantly improved performance is observed when increasing the embedding dimension from 20 to 50. However, when the embedding dimension exceeds 50, the model's performance starts to decline. Our research suggests that smaller embedding dimensions may reduce model performance, while overly large dimensions may lead to overfitting.

(3) The MHCLR model proposed in this paper outperforms other recommendation system models on three datasets. The performance of the MHCLR algorithm is consistently superior to that of BPR, GraphRec, LightGCN, and SGL algorithms, indicating that hypergraph-based models have better expressive capabilities compared to graph-based models. Although the DHCF algorithm based on hypergraphs does not perform well in recommendation, this may be due to the inappropriate construction of hyperedges in the model, resulting in high matrix density or suboptimal experimental results. Combining the above analysis, the strong performance of MHCLR can be attributed to three main factors. First, the assistance of multi-source heterogeneous networks in the recommendation system, utilizing hypergraphs to establish high-order relationships, taking into account social relationships, sequence relationships and so on. Second, on the foundation of capturing complex high-order user relationships with Hyper-GAT, we focus on the density information relationships among different hyperedges, fully leveraging the implicit information within multi-source heterogeneous hypergraphs. Additionally, we design a multi-view cross-contrast learning module that combines with hypergraph learning, compensating for potential multivariate modeling information loss during the alignment process and further enhancing accuracy.

In the ablation experiments, as shown in Figure 8 and Table 4, we can draw the following conclusions:

(1) Replacing the SD-HGAT module in our algorithm with traditional HGNN and Hyper-GAT modules results in a decrease in algorithm performance across all three datasets. This indicates that the SD-HGAT module has significant potential in extracting graph features.

(2) When the contrastive learning module is not used to enhance multivariate modeling accuracy, and only the SD-HGAT module is employed for recommendation, the performance of the MHCLR algorithm slightly decreases. Additionally, when recommending using direct fusion of multiple views, the performance is lower compared to cross-view fusion. This is because cross-view fusion handles heterogeneity among different views effectively, utilizing the characteristics of each view to a greater extent. Therefore, the multi-view cross-contrast learning module proposed in this paper has a positive effect on preserving heterogeneity and improving accuracy in modeling multivariate relationships.

V. CONCLUSION

This paper mainly focuses on enhancing the performance of recommender systems through multi-view contrastive learning guided by multi-source heterogeneous hypergraphs. We propose the MHCLR algorithm, which includes a hypergraph attention network learning module with hypergraph structures and density information, as well as a multi-view encoding contrastive mode. Specifically, by using hypergraphs to fuse multi-source heterogeneous graph information, we can capture high-order correlations between items or users, achieve high-quality information integration, and model high-order feature interactions. The spatial density hypergraph attention network module is introduced to fully exploit node density information and composite hyperedge density information in multi-source heterogeneous hypergraphs, enhancing user and item feature representations. In terms of contrastive learning, we enhance the accuracy of modeling multi-variable specific relationships at different levels by fusing multi-view information and hypergraph contrasts, reducing redundancy. We demonstrated the effectiveness of our approach in terms of Precision, Recall, and NDCG metrics through experiments, achieving significant improvements in recommendation accuracy.

While the hypergraph data structure is a relatively new development in recommender systems and has a positive impact on performance improvement, hypergraph neural networks still face some of the shortcomings of traditional graph neural networks, such as lack of interpretability and inability to directly model causal relationships. Future research directions should focus on discovering more fine-grained causal relationships. This can be achieved by establishing counterfactual learning in hypergraphs to eliminate false factors caused by confounding factors, uncovering the underlying reasons for user feedback behaviors, and accurately capturing true user preferences hidden in the noisy feedback data to further improve recommendation effectiveness.

REFERENCES

[1] X. Zhang, H. Ma, Z. Gao, Z. Li, and L. Chang, "Exploiting cross-session information for knowledge-aware session-based recommendation via graph attention networks," *Int. J. Intell. Syst.*, vol. 37, no. 10, pp. 7614–7637, Oct. 2022.

[2] M. Ding, X. Lin, B. Zeng, and Y. Chai, "Hypergraph neural networks with attention mechanism for session-based recommendation," *J. Phys., Conf. Ser.*, vol. 2082, no. 1, Nov. 2021, Art. no. 012007.

[3] Z. Zhu, S. Wang, F. Wang, and Z. Tu, "Recommendation networks of homogeneous products on an e-commerce platform: Measurement and competition effects," *Expert Syst. Appl.*, vol. 201, Sep. 2022, Art. no. 117128.

[4] Z. Wang, "English news text recommendation method based on hypergraph random walk label expansion," *Comput. Intell. Neurosci.*, vol. 2022, Mar. 2022, Art. no. 1880114.

[5] L. Danyang and G. Mingxin, "Music recommendation method based on multi-source information fusion," *Data Anal. Knowl. Discovery*, vol. 5, no. 2, pp. 94–105, 2021.

[6] H. Liu, C. Zheng, D. Li, Z. Zhang, K. Lin, X. Shen, N. N. Xiong, and J. Wang, "Multi-perspective social recommendation method with graph representation learning," *Neurocomputing*, vol. 468, pp. 469–481, Jan. 2022.

[7] C. Feng, J. Liang, P. Song, and Z. Wang, "A fusion collaborative filtering method for sparse data in recommender systems," *Inf. Sci.*, vol. 521, pp. 365–379, Jun. 2020.

[8] F. J. Tey, T.-Y. Wu, C.-L. Lin, and J.-L. Chen, "Accuracy improvements for cold-start recommendation problem using indirect relations in social networks," *J. Big Data*, vol. 8, no. 1, pp. 1–18, Dec. 2021.

[9] G. Zhou, N. Mou, Y. Fan, Q. Pi, W. Bian, C. Zhou, X. Zhu, and K. Gai, "Deep interest evolution network for click-through rate prediction," in *Proc. AAAI*, 2019, pp. 5941–5948.

[10] S. Cheng, B. Zhang, G. Zou, M. Huang, and Z. Zhang, "Friend recommendation in social networks based on multi-source information fusion," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 5, pp. 1003–1024, May 2019.

[11] Y. Guo, C. Yin, M. Li, X. Ren, and P. Liu, "Mobile e-commerce recommendation system based on multi-source information fusion for sustainable e-business," *Sustainability*, vol. 10, no. 2, p. 147, Jan. 2018.

[12] L. Zhang, Y. Xie, L. Xidao, and X. Zhang, "Multi-source heterogeneous data fusion," in *Proc. Int. Conf. Artif. Intell. Big Data (ICAIBD)*, May 2018, pp. 47–51.

[13] X. Zhao, Y. Jia, A. Li, R. Jiang, and Y. Song, "Multi-source knowledge fusion: A survey," *World Wide Web*, vol. 23, no. 4, pp. 2567–2592, Jul. 2020.

[14] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. Orgun, "Sequential recommender systems: Challenges, progress and prospects," 2019, *arXiv:2001.04830*.

[15] Y. Li, J. Liu, and J. Ren, "Social recommendation model based on user interaction in complex social networks," *PLoS ONE*, vol. 14, no. 7, Jul. 2019, Art. no. e0218957.

[16] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, Feb. 2019.

[17] Y. Yang, C. Huang, L. Xia, and C. Huang, "Knowledge graph self-supervised rationalization for recommendation," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2023, pp. 3046–3056.

[18] D. Peng and S. Zhang, "GC-HGNN: A global-context supported hypergraph neural network for enhancing session-based recommendation," *Electron. Commerce Res. Appl.*, vol. 52, pp. 101–129, Mar. 2022.

[19] J. Wang, K. Ding, L. Hong, H. Liu, and J. Caverlee, "Next-item recommendation with sequential hypergraphs," in *Proc. SIGIR*, 2020, pp. 1101–1110.

[20] Y. Gao, Z. Zhang, H. Lin, X. Zhao, S. Du, and C. Zou, "Hypergraph learning: Methods and practices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2548–2566, May 2022.

[21] L. Zhang, J. Guo, J. Wang, J. Wang, S. Li, and C. Zhang, "Hypergraph and uncertain hypergraph representation learning theory and methods," *Mathematics*, vol. 10, no. 11, p. 1921, Jun. 2022.

[22] V. L. Gatta, V. Moscato, M. Pennone, M. Postiglione, and G. Sperlí, "Music recommendation via hypergraph embedding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 7887–7899, Oct. 2023.

[23] S. Agarwal, K. Branson, and S. Belongie, "Higher order learning with graphs," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 17–24.

[24] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3558–3565.

[25] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar, "HyperGCN: A new method for training graph convolutional networks on hypergraphs," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 1511–1522.

[26] K. Ding, J. Wang, J. Li, D. Li, and H. Liu, "Be more with less: Hypergraph attention networks for inductive text classification," 2020, *arXiv:2011.00387*.

[27] J. Liao, Q. Tao, and J. Yan, "Density-aware hyper-graph neural networks for graph-based semi-supervised node classification," 2022, *arXiv:2201.11511*.

[28] J. J. Whang, R. Du, S. Jung, G. Lee, B. Drake, Q. Liu, S. Kang, and H. Park, "MEGA: Multi-view semi-supervised clustering of hypergraphs," *Proc. VLDB Endowment*, vol. 13, no. 5, pp. 698–711, Jan. 2020.

[29] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 18661–18673.

[30] M. Chen, C. Huang, L. Xia, W. Wei, Y. Xu, and R. Luo, "Heterogeneous graph contrastive learning for recommendation," in *Proc. 16th ACM Int. Conf. Web Search Data Mining*, Feb. 2023, pp. 544–552.

[31] Y. Wu, R. Xie, Y. Zhu, X. Ao, X. Chen, X. Zhang, F. Zhuang, L. Lin, and Q. He, "Multi-view multi-behavior contrastive learning in recommendation," in *Proc. Int. Conf. Database Syst. Adv. Appl.* Cham, Switzerland: Springer, 2022, pp. 166–182.

[32] A. Khelloufi, H. Ning, S. Dhelim, T. Qiu, J. Ma, R. Huang, and L. Atzori, "A social-relationships-based service recommendation system for SIoT devices," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1859–1870, Feb. 2021.

[33] L. Sun, "POI recommendation method based on multi-source information fusion using deep learning in location-based social networks," *J. Inf. Process. Syst.*, vol. 17, no. 2, pp. 352–368, 2021.

[34] X. Yu, "Global multi-source information fusion management and deep learning optimization for tourism: Personalized location-based service," *J. Organizational End User Comput.*, vol. 34, no. 3, pp. 1–21, Dec. 2021.

[35] Y. Wang, J. Wen, W. Zhou, X. Wang, Q. Wu, and B. Tao, "A cloud service recommendation method based on extended multi-source information fusion," *Concurrency Comput., Pract. Exper.*, vol. 34, no. 10, p. e6826, May 2022.

[36] D. Wang, X. Zhang, D. Yu, G. Xu, and S. Deng, "CAME: Content- and context-aware music embedding for recommendation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 1375–1388, Mar. 2021.

[37] R. Sun, X. Cao, Y. Zhao, J. Wan, K. Zhou, F. Zhang, Z. Wang, and K. Zheng, "Multi-modal knowledge graphs for recommender systems," in *Proc. CIKM*, 2020, pp. 1405–1414.

[38] Y. Zhu, Z. Guan, S. Tan, H. Liu, D. Cai, and X. He, "Heterogeneous hypergraph embedding for document recommendation," *Neurocomputing*, vol. 216, pp. 150–162, Dec. 2016.

[39] S. Tan, J. Bu, C. Chen, B. Xu, C. Wang, and X. He, "Using rich social media information for music recommendation via hypergraph model," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 7S, no. 1, pp. 1–22, Oct. 2011.

[40] T. Jin, L. Cao, B. Zhang, X. Sun, C. Deng, and R. Ji, "Hypergraph induced convolutional manifold networks," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2670–2676.

[41] R. Zhang, Y. Zou, and J. Ma, "Hyper-SAGNN: A self-attention based graph neural network for hypergraphs," 2019, *arXiv:1911.02613*.

[42] M. Wang, X. Liu, and X. Wu, "Visual classification by ℓ_1-hypergraph modeling," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 9, pp. 2564–2574, Sep. 2015.

[43] Y. Fang and Y. Zheng, "Metric learning based on attribute hypergraph," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3440–3444.

[44] Y. Gao, Y. Feng, S. Ji, and R. Ji, "HGNN+: General hypergraph neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3181–3199, Mar. 2023.

[45] Z. Xia, W. Zhang, and Z. Weng, "Social recommendation system based on hypergraph attention network," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–12, Nov. 2021.

[46] L. He, H. Chen, D. Wang, S. Jameel, P. Yu, and G. Xu, "Click-through rate prediction with multi-modal hypergraphs," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, pp. 690–699.

[47] E.-S. Kim, W. Y. Kang, K.-W. On, Y.-J. Heo, and B.-T. Zhang, "Hypergraph attention networks for multimodal learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14569–14578.

[48] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[49] T. Carletti, F. Battiston, G. Cencetti, and D. Fanelli, "Random walks on hypergraphs," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 101, no. 2, Feb. 2020, Art. no. 022308.

[50] S. Bai, F. Zhang, and P. H. Torr, "Hypergraph convolution and hypergraph attention," *Pattern Recognit.*, vol. 110, Feb. 2021, Art. no. 107637.

[51] J. Yi and J. Park, "Hypergraph convolutional recurrent neural network," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 3366–3376.

[52] G. Karantaidis, I. Sarridis, and C. Kotropoulos, "Adaptive hypergraph learning with multi-stage optimizations for image and tag recommendation," *Signal Process., Image Commun.*, vol. 97, Sep. 2021, Art. no. 116367.

[53] Z. Wang, J. Chen, F. E. Rosas, and T. Zhu, "A hypergraph-based framework for personalized recommendations via user preference and dynamics clustering," *Expert Syst. Appl.*, vol. 204, Oct. 2022, Art. no. 117552.

[54] S. Ji, Y. Feng, R. Ji, X. Zhao, W. Tang, and Y. Gao, "Dual channel hypergraph collaborative filtering," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 2020–2029.

[55] H. Yuan, J. Yang, and J. Huang, "Improving hypergraph convolution network collaborative filtering with feature crossing and contrastive learning," *Int. J. Speech Technol.*, vol. 52, no. 9, pp. 10220–10233, Jul. 2022.

[56] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, "GCC: Graph contrastive coding for graph neural network pretraining," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1150–1160.

[57] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 4116–4126.

[58] Y. You, T. Chen, Y. Shen, and Z. Wang, "Graph contrastive learning automated," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12121–12132.

[59] Y. Tao, M. Gao, J. Yu, Z. Wang, Q. Xiong, and X. Wang, "Predictive and contrastive: Dual-auxiliary learning for recommendation," *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 5, pp. 2254–2265, Oct. 2023.

[60] J. Shuai, K. Zhang, L. Wu, P. Sun, R. Hong, M. Wang, and Y. Li, "A review-aware graph contrastive learning framework for recommendation," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2022, pp. 1283–1293.

[61] J. Yu, H. Yin, J. Li, Q. Wang, N. Q. V. Hung, and X. Zhang, "Self-supervised multi-channel hypergraph convolutional network for social recommendation," in *Proc. Web Conf.*, Apr. 2021, pp. 413–424.

[62] L. Xia, C. Huang, Y. Xu, J. Zhao, D. Yin, and J. Huang, "Hypergraph contrastive collaborative filtering," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2022, pp. 70–79.

[63] D. Zou, W. Wei, X.-L. Mao, Z. Wang, M. Qiu, F. Zhu, and X. Cao, "Multi-level cross-view contrastive learning for knowledge-aware recommender system," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2022, pp. 1358–1368.

[64] T. Wei, Y. You, T. Chen, Y. Shen, J. He, and Z. Wang, "Augmentations in hypergraph contrastive learning: Fabricated and generative," 2022, *arXiv:2210.03801*.

[65] D. Cai, C. Sun, M. Song, B. Zhang, S. Hong, and H. Li, "Hypergraph contrastive learning for electronic health records," in *Proc. SDM*, 2022, pp. 127–135.

[66] Y. Sun, D. Zhu, H. Du, and Z. Tian, "Motifs-based recommender system via hypergraph convolution and contrastive learning," *Neurocomputing*, vol. 512, pp. 323–338, Nov. 2022.

[67] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 639–648.

[68] R. Wang, R. Shivanna, D. Cheng, S. Jain, D. Lin, L. Hong, and E. Chi, "DCN V2: Improved deep & cross network and practical lessons for web-scale learning to rank systems," in *Proc. Web Conf.*, Apr. 2021, pp. 1785–1797.

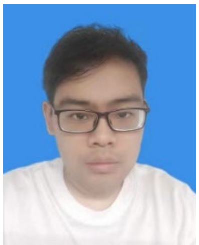
[69] H. Yin, Q. Wang, K. Zheng, Z. Li, J. Yang, and X. Zhou, "Social influence-based group representation learning for group recommendation," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Apr. 2019, pp. 566–577.

[70] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," 2012, *arXiv:1205.2618*.

[71] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *Proc. World Wide Web Conf.*, May 2019, pp. 417–426.

[72] X. Liu, "Global self-supervised graph learning for recommendation," in *Proc. IEEE 3rd Int. Conf. Electr. Eng., Big Data Algorithms (EEBDA)*, Feb. 2024, pp. 726–735.

SHANSHAN WAN received the Ph.D. degree from the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China, in 2018. She is currently an Associate Professor with the School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, China. Her current research interests include image processing, data mining, natural language processing, intelligent tutoring systems, and smart building.



JIAQI DING received the B.S. degree in engineering from the School of Automation and Electrical Engineering, Shenyang Ligong University. He is currently pursuing the M.S. degree with Beijing University of Civil Engineering and Architecture, China. His current research interests include data mining and natural language processing.