

Self-supervised Recommendation with Cross-channel Matching Representation and Hierarchical Contrastive Learning.

DONGJIE ZHU, YUNDONG SUN*, HAIWEN DU, and ZHAOSHUO TIAN, Harbin Institute of Technology, China

Recently, leveraging different channels to model social semantic information and using self-supervised learning tasks to boost recommendation performance has been proven to be a very promising work. However, how to deeply dig out the relationship between different channels and make full use of it while maintaining the uniqueness of each channel is a problem that has not been well studied and resolved in this field. Under such circumstances, this paper explores and verifies the deficiency of directly constructing contrastive learning tasks on different channels with practical experiments and proposes the scheme of interactive modeling and matching representation across different channels. This is the first attempt in the field of recommender systems, we believe the insight of this paper is inspirational to future self-supervised learning research based on multi-channel information. To solve this problem, we propose a cross-channel matching representation model based on attentive interaction, which realizes efficient modeling of the relationship between cross-channel information. Based on this, we also propose a hierarchical self-supervised learning model, which realize two levels of self-supervised learning within and between channels, which improves the ability of self-supervised tasks to autonomously mine different levels of potential information. We have conducted abundant experiments, and various metrics on multiple public datasets show that the method proposed in this paper has a significant improvement compared with the state-of-the-art methods, no matter in the general or cold-start scenario. And in the experiment of model variant analysis, the benefits of the cross-channel matching representation model and the hierarchical self-supervised model proposed in this paper is also fully verified.

CCS Concepts: • **Information systems** → **Recommender systems**; • **Computing methodologies** → **Neural networks**.

Additional Key Words and Phrases: Self-supervised learning, Graph neural networks, Hypergraph, Contrastive learning

ACM Reference Format:

Dongjie Zhu, Yundong Sun, Haiwen Du, and Zhaoshuo Tian. 2021. Self-supervised Recommendation with Cross-channel Matching Representation and Hierarchical Contrastive Learning. . *J. ACM* 37, 4, Article 111 (August 2021), 23 pages. <https://doi.org/10.1145/1122445.1122456>

*Yundong Sun is the corresponding author.

Authors' address: Dongjie Zhu, zhudongjie@hit.edu.cn; Yundong Sun, hitffmy@163.com; Haiwen Du, duhaiwen@126.com; Zhaoshuo Tian, tianzhaoshuo@126.com, Harbin Institute of Technology, No.92,Xidazhi Street,Nangang District, Harbin, Heilongjiang, China, 150001.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

0004-5411/2021/8-ART111 \$15.00

<https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

With the development of network technology and the popularization of mobile smart devices, especially the emergence of micro-video sharing platforms, such as Tiktok¹ and Kwai², information has shown explosive growth. Enormous information is stored on the network, but it is more and more difficult for us to obtain valuable information for ourselves, which brings greater challenges to the recommender systems[34].

In recent years, the graph neural networks(GNNs) have achieved unprecedented success in various tasks, such as node classification, link prediction, and graph classification with its strong ability to extract graph data structure, node attributes, and relationships between nodes[40, 44, 48]. At the same time, the application of GNNs in recommender systems has also become a hot topic. Some studies construct the bipartite graphs from interactive information between users and items and utilize GNNs to dig out multi-hop relationships between users and users or items and items [15, 26], breaking restrictions of the traditional collaborative filtering paradigms on capturing higher-order relationships. In addition, some methods add knowledge graph [2, 31] and social information[47] as supplementary information to the interaction graph, which further improves the performance of the recommender systems.

More information means more complex interactions. The interactions can be different patterns of social relationships between users[7, 42], or between users and items[43]. Obviously, these complex and high-order interactions can provide richer information and help improve recommendation performance. Although stacking multi-layer GNNs can capture the high-order relationship of multiple hops between nodes, it cannot capture the high-order relationship with specific patterns. The emergence of hypergraph neural networks (Hyper-GNNs) solved this problem. The hypergraph connects nodes with a specific relationship pattern through an edge. This natural structure makes it possible to mine specific interaction patterns between multiple nodes[7, 13].

Although the introduction of Hyper-GNNs and various interactive information with specific patterns has greatly improved the performance of the recommendation model, this performance improvement is guaranteed by enormous interactive information. For new users or items in the cold-start scenario, the recommendation performance is poor due to the lack of interactive information. At the same time, simply aggregating interaction information with specific patterns through different channels will also lose the characteristics of different channel data. Self-Supervised Learning (SSL) can realize label-free learning with the help of self-supervised tasks, and it can also dig out potentially valuable information autonomously, alleviating the cold-start problem to some extent. More importantly, it can fully mine and utilize the characteristics of different channel information to promote the improvement of recommendation performance. In the field of Computer Vision (CV) and Natural Language Processing (NLP), various SSL methods[3, 5, 11] have been used to solve the deficiencies of data labels and have achieved great success. However, due to the continuity of node attributes in the interaction graph and the complex relationship between nodes, it is difficult to directly apply the data augmentation methods in the field of CV and NLP to recommendation[35].

Researchers have also conducted a lot of studies on SSL in the field of recommender systems. The existing methods mainly focus on generating different local substructures by randomly removing edges, sampling nodes[35, 41] to obtain training samples for contrastive learning, which is beneficial to capturing the graph structure with various patterns. However, there is rich semantic information of interactions in the recommender systems, such as various social relationships between users, and the association relationships between items and items. It is difficult to effectively mine the semantic information of multiple interaction patterns by simply using random methods for data

¹<https://www.tiktok.com/>

²<https://www.kwai.com/>

augmentation. Different channels can not only provide richer semantic information, which is beneficial to improve performance, but also provide new possibilities for self-supervision. Some of the latest methods[43] have been aware of the importance of semantic information from different interaction patterns and use different channels to capture the information of different patterns and construct self-supervised tasks in each channel. However, the existing methods cannot make full use of the relationship between different channels to construct self-supervised tasks. Simply constructing contrastive learning tasks for the data from different channels by some methods (e.g., maximum mutual information.) will make the data of each channel extremely homogeneous and lose their uniqueness (We also call it homogeneity, we proved this in exploratory experiments, see Preliminaries 2.3.). Therefore, we need a new method to efficiently model the relationship between the interactive information of different channels to get cross-channel information that can be used to establish self-supervised tasks, so recommendation performance can be boosted. At the same time, it reduces the dependence on data labels and data volume and further improves performance in cold-start scenarios.

Based on the above analysis, we proposed a self-supervised recommendation framework with Cross-channel Matching representation and Hierarchical Co-contrastive learning, which is also called CMHC. First, inspired by MHCN[43], the data of different channels in the interaction graph is extracted, including Social-channel, Purchase-channel, and Joint-channel, and the Hyper-GNN is leveraged to encode information of each channel. On this basis, we innovatively propose a cross-channel matching representation model, which constructs first-order ego-networks of the nodes under each channel and realizes the information transition between channels based on the proposed Attentive-Matching representation learning model. More importantly, to solve the data homogeneity problem of each channel (Preliminaries 2.3), a hierarchical SSL model based on cross-channel matching representation is proposed, and a self-supervised task is established for matching representations under different channels.

The contributions of this paper can be summarized as follows:

1. We make explorations and experimentally verify the deficiency of simply constructing contrastive learning tasks for the data from different channels and propose the scheme of interactive modeling and matching representation across different channels. As far as we know, this is the first attempt in the field of recommender systems, which can help to facilitate subsequent self-supervised learning research based on multi-channel information.
2. We propose a self-supervised recommendation framework with Cross-channel Matching representation and Hierarchical Co-contrastive learning, which is also called CMHC. Specifically, we propose a cross-channel matching representation model based on attentive interaction, which realizes efficient modeling of the relationship between cross-channel information. Based on this, we also proposed a hierarchical self-supervised model, which realized two levels of self-supervised learning within and between channels and improved the ability of self-supervised tasks to autonomously mine different levels of potential information.
3. We have conducted abundant experiments on various real datasets, and the results show that CMHC proposed in this paper outperforms the state-of-the-art methods by a big margin, no matter in the general or cold-start scenario. And in the model variant analysis experiment, the benefits of the cross-channel matching representation model and the hierarchical self-supervised model proposed in this paper is also fully verified.

The rest of this paper is arranged as follows: Chapter 2 will introduce some definitions and basic technologies involved in this paper, such as hypergraph and hypergraph neural networks, and analyze our exploration experiments on self-supervised contrastive learning on multi-channel information. We will describe and analyze our method in Chapter 3 detailly. Chapter 4 is the section

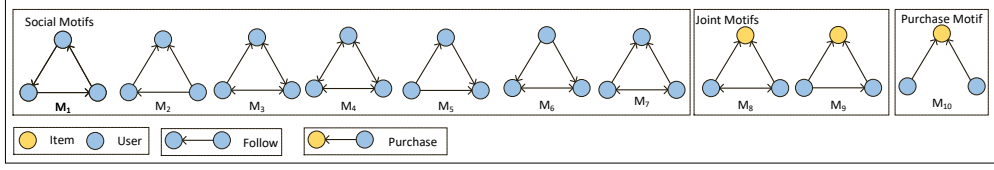


Fig. 1. The triangle relationships extracted in this paper.

of the experiments. We will verify the method proposed in this paper, including performance comparison experiments, variant experiments, and stability experiments. Chapter 5 will introduce and analyze the research work related to this paper, including hypergraph neural network, self-supervised learning, and self-supervised graph learning for recommendation. Chapter 6 will conclude this paper and some future works.

2 PRELIMINARIES

2.1 Hypergraph Definition

For a general graph, an edge can only connect two nodes, so that it can only describe a pairwise relationship. But in real scenarios, many relationships are not limited to two nodes, such as the stable triangular relationship in social networks[42]. The hypergraph breaks this limitation. The hypergraph evolves an edge into a hyperedge, and a hyperedge can connect multiple nodes at the same time, which is also the core idea of the hypergraph[7]. A hypergraph can be defined as $G = (V, E)$, where V represents the set of nodes, and E represents the set of hyperedges in the graph. The adjacency relationship between nodes can be represented as matrix $A \in \{0, 1\}^{|V| \times |E|}$, where each element indicates whether hyperedge e contains node v :

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

Like a general graph, the degree of a node in hypergraph can be expressed by the number of hyperedges that contain the node, that is $d(v) = \sum_{e \in E} a(v, e)$. In addition, because a hyperedge is no longer limited to connecting two nodes, the degree of the hyperedge can be additionally defined as the number of nodes connected by it, that is $d(e) = \sum_{v \in V} a(v, e)$. Therefore, the diagonal matrices $D_e \in \mathbb{R}^{|E| \times |E|}$ and $D_v \in \mathbb{R}^{|V| \times |V|}$ formed by $d(e)$ and $d(v)$ represent the degree matrix of the hyperedges and nodes in hypergraph G , respectively.

2.2 Extraction and Encoding of Information with Different Interaction Patterns

Birds of a feather flock together. Closely connected people are most likely to have the same interests. At present, with the popularity of various social platforms and the accumulation of massive data, it has become a trend to integrate users' social network information into the recommender systems[42, 43]. The fusion of social network information and User-Item interactive information makes the relationship between user-user and user-item richer and no longer limited to pairwise relationships, which brings both advantages and challenges to relationship mining. How to extract and encode information with various patterns is not the focus of this paper. We follow the relevant content in MHCN[43].

First, we extract the relationship of the three patterns, namely Purchase Motif, Joint Motif, and Social Motif, as shown in Figure 1. A large number of studies have proven that triangles are extremely important in social relationships[1, 42], among which Social Motif (M_1 - M_7) covers the

most important and stable triangle social relationships³. Joint Motif (M_8 and M_9) means that two users have a social relationship and bought the same product. Purchase Motif (M_{10}) means that two users who do not have social relationships have purchased the same product. Although they have no social relationships, their demands or interests are similar. This is a very important potential relationship for recommendations.

Table 1. The matrix computation of differnet motifs.

Motif	Matrix Computation	A_{M_i}
M_1	$C = (UU) \odot U^\top$	$C + C^\top$
M_2	$C = (BU) \odot U^\top + (UB) \odot U^\top + (UU) \odot B$	$C + C^\top$
M_3	$C = (BB) \odot U + (BU) \odot B + (U \cdot B) \odot B$	$C + C^\top$
M_4	$C = (BB) \odot B$	C
M_5	$C = (UU) \odot U + (UU^\top) \odot U + (U^\top U) \odot U$	$C + C^\top$
M_6	$C = (UB) \odot U + (BU^\top) \odot U^\top + (U^\top U) \odot B$	C
M_7	$C = (U^\top B) \odot U^\top + (BU) \odot U + (U^\top U) \odot B$	C
M_8	$C = (RR^\top) \odot B$	C
M_9	$C = (RR^\top) \odot U$	$C + C^\top$
M_{10}	$C = RR^\top$	C

The corresponding adjacency matrix extraction method is shown in Table 1. After obtaining the adjacency matrix A_{M_K} under each motif, then the adjacency matrix of Social Motifs is $A_s = \sum_{k=1}^7 A_{M_k}$; the adjacency matrix of Joint Motifs is $A_j = A_{M_8} + A_{M_9}$; the adjacency matrix of Purchase Motifs is $A_p = A_{M_{10}} - A_j$.

After obtaining the adjacency matrix in the three channels, the hypergraph convolutional neural network proposed in [7] can be used to encode information of each channel:

$$H_c^{(l+1)} = \hat{D}_c^{-1} A_c H_c^{(l)} \quad (2)$$

where $D_c \in \mathbb{R}^{n \times n}$ is the degree matrix of A_c .

2.3 Empirical Explorations of Self-supervised Learning on Cross-Channel Information

In this section, we explore how to leverage different channels of data with different interaction patterns to perform self-supervised learning. An intuitive idea is that after obtaining the representation of each channel, the contrastive loss function is used to make the representation between the different channels of the positive samples closer, and the representation of the negative samples more distant. Therefore, we add Triplet-SSL loss[6] and InfoNCE-SSL loss[9] to the $S^2 - MHCN$ [43] to build the contrastive learning on multi-channel information directly, formatting the model of $S^2 - MHCN_{+Triplet}$ and $S^2 - MHCN_{+InfoNCE}$. We keep the other hyperparameters unchanged to verify the performance of the model variants on different data. The exploration results are shown in Figure 2.

It can be seen from Figure 2 that the performance of $S^2 - MHCN_{+Triplet}$ and $S^2 - MHCN_{+InfoNCE}$ on all metrics has a consistent decline compared with $S^2 - MHCN$, which indicates that building the contrastive learning on multi-channel information directly is not a satisfactory scheme. This is in line with the original assumption of this paper: building the contrastive learning on multi-channel information directly will make the data of each channel extremely homogeneous and lose their uniqueness. The homogeneity problem not only fails to improve but also deteriorates the performance of the recommendation. Therefore, a new method is needed to efficiently model

³We have also explored other triangle relations except for M_1 - M_7 but found that adding these triangle relations does not improve the performance.

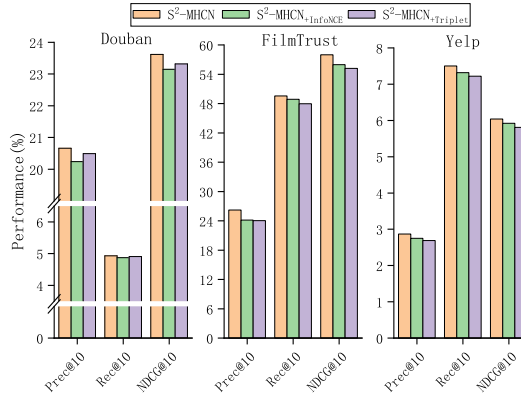


Fig. 2. The performance exploration results of building the contrastive learning on multi-channel information directly.

the relationships of different channels while maintaining their characteristics, to maximize the advantages of SSL, which is the key problem this paper to solve.

3 THE PROPOSED METHOD

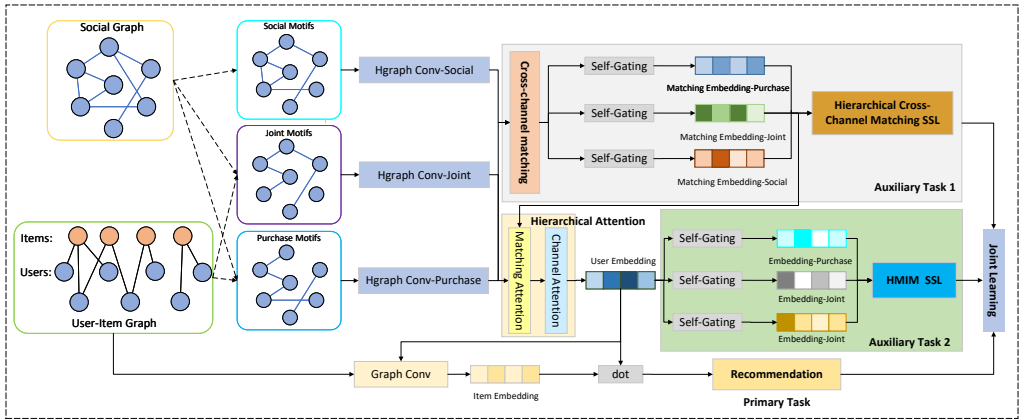


Fig. 3. Schematic diagram of the CMHC framework(one layer) proposed in this paper.

This section will detail our proposed CMHC framework, its overall structure diagram is shown in Figure 3. First, we construct three channels of information with various interaction patterns based on Social Graph and User-Item Graph, namely Purchase Motif, Joint Motif, and Social Motif[23, 43] (Preliminaries 2.2). Then, the Hyper-GNN is used to encode information of each channel to obtain the user representation under each channel. For Item, the GNN is performed on User-Item Graph to obtain its representation (Preliminaries 2.2). Finally, based on the user embedding representations of each channel and the item representations, we build joint learning between the auxiliary task of the hierarchical SSL and the main task of recommendation. Cross-channel matching representation learning (section 3.1), cross-channel hierarchical co-contrastive learning model (section 3.2), and user representation fusion based on hierarchical attention (section 3.3) are the core of this paper.

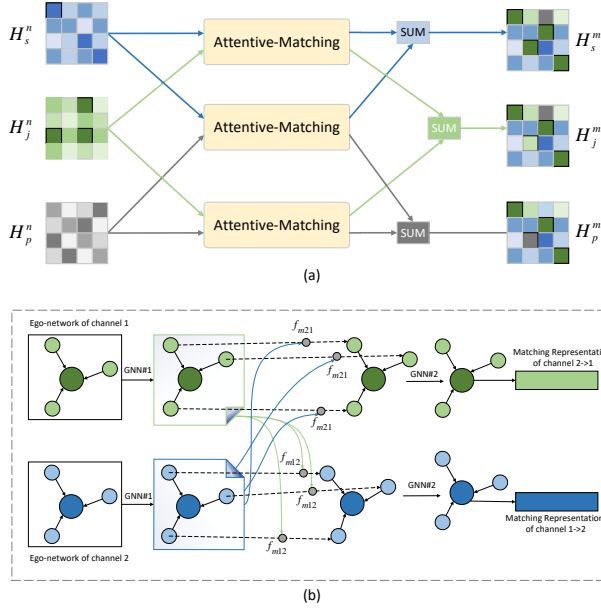


Fig. 4. (a) Framework of Cross-channel Matching representation learning and (b) details of Attentive-Matching model.

3.1 Cross-channel Matching Representation Learning

3.1.1 Framework of Cross-channel Matching Representation Learning. We use the three channels as examples to illustrate and analyze the cross-channel matching representation framework proposed in this paper. The overall diagram of the architecture is shown in Figure 4 (a). First, for the data of any two channels, use the cross-channel representation learning model (Attentive-Matching) proposed in this article to obtain the transitional matching representation from channel 2 to channel 1 or from channel 1 to channel 2. Then the transitional matching representations from the other two channels to the current channel are summed to obtain the cross-channel transitional representation of the current channel. For example, the cross-channel matching representation of social -channel can be represented as the follows:

$$H_s^m = Attm(H_s^n, H_j^n) + Attm(H_s^n, H_p^n) \quad (3)$$

where H_s^n, H_j^n, H_p^n is the user representation under each channel obtained by equation (2), $Attm(\cdot)$ is the Attentive-Matching model we proposed in section 4.1.2. We extend it to a more general case (there are L channels), and the matching representation of each channel is calculated as follows:

$$H_i^m = \sum_{0 < l \leq L, l \neq i} Attm(H_i^n, H_l^n), i = 1, 2, \dots, L \quad (4)$$

where H_i^n is the user representation under channel i , and H_i^m is the cross-channel matching representation of the user under channel i .

3.1.2 Attentive-Matching Model. To obtain the transitional matching representation between two channels, this paper proposes an Attentive-Matching cross-channel representation learning model. First, we use GCN#1[16] to gather information on the ego-network under each channel to obtain the ego-network representations g_i^1 and g_j^2 of the node i and j .

On this basis, the nodes of ego-network under other channels are used to perform cross-channel transitional representation modeling for the nodes of the current channel. Specifically, we firstly calculate the cosine similarity between g_i^1 and g_j^2 , as shown in equation (5).

$$\alpha_{i,j} = \cos \text{ine}(g_i^1, g_j^2), j \in 1, \dots, |G_2| \quad (5)$$

Then, we perform a weighted summation on the representation of all nodes in G_2 to obtain the transitional representation of the nodes in G_1 , as shown in equation (6).

$$g_i^{-1} = \frac{\sum_{j=1}^{|G_2|} \alpha_{i,j} * g_j^2}{\sum_{j=1}^{|G_2|} \alpha_{i,j}} \quad (6)$$

Finally, we utilize representation g_i^1 and transitional representation g_i^{-1} to obtain the transitional matching representation of node i from channel 2 to channel 1, as shown in equation(7).

$$h_i^m = f_m(g_i^1, g_i^{-1}) \quad (7)$$

where f_m is a multi-view cosine matching function that compares two vectors of v_1 and v_2 :

$$h^m = f_m(v_1, v_2, W) \quad (8)$$

where $W \in \mathbb{R}^{l \times d}$ is a trainable parameter, l is a multi-view parameter, and W_k represents one row of W , which represents a single view. d is the dimension of the input vectors, and $m = [m_1, m_2, \dots, m_k, \dots, m_l]$ is a l -dimensional matching vector. In special cases, if $l = 1$, it is the common vector cosine similarity of the input vectors:

$$m_1 = \text{cosine}(W_1 * v_1, W_1 * v_2) \quad (9)$$

To better capture the local structure information of the nodes, GCN#2[16] is used again to gather the ego-network information to obtain the final cross-channel matching representation under each channel. Similarly, we apply the same process to the nodes in G_2 .

3.2 Hierarchical Self-supervised Learning based on Cross-channel Matching Representation.

In section 3.1, we obtained the transitional matching representation between different channel data. This kind of matching representation data contains the association relationship between different channels, which becomes the basis of applying contrastive learning among channels. At the same time, like general channel information, the matching representations under each channel still maintain different distribution, which contains unique characteristics. To preserve the uniqueness of the respective channels and make full use of the correlation between the channels of the matching representation, this paper proposes a hierarchical self-supervised learning model based on cross-channel matching representation. The diagram of the model is shown in Figure 5.

First, we need to build a contrastive learning task on representations of multi-channels. The existing commonly used objective functions include the maximum mutual information model[28] and the InfoNCE model[10]. In the end, it can correctly capture the similarity representation of instances between different channels.

Specifically, in each calculation process, one positive sample(k_+) and K negative samples (k_0, \dots, k_K) in the queue are selected for calculation, and InfoNCE[10] can be used to perform the following calculations:

$$\ell = -\log \frac{\exp(h_q^\top h_{k_+}^m / \tau)}{\sum_{i=0}^K \exp(h_q^\top h_{k_i}^m / \tau)} \quad (10)$$

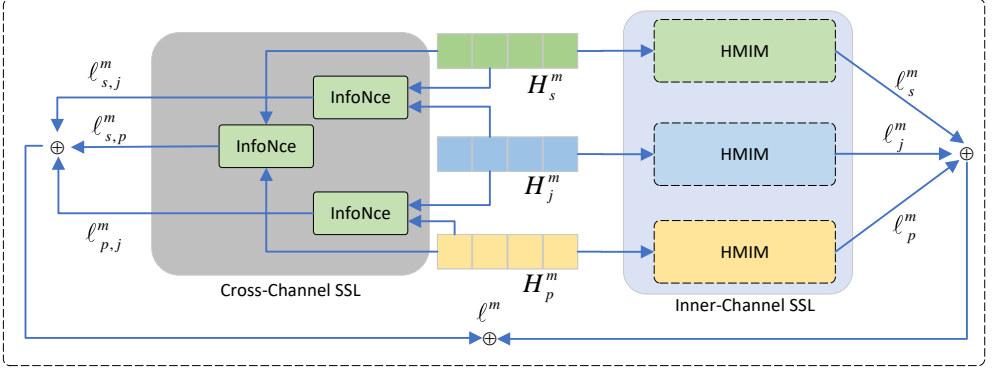


Fig. 5. (a) Framework of Cross-channel Matching representation learning and (b) Details of Attentive-Matching model.

where h_q is the representation of the query node, here it is the matching representation of node q under a certain channel obtained in section 3.1. $h_{k_i}^m$ is the matching representation of the positive sample, and $h_{k_i}^m$ is the matching representation of the i^{th} negative sample. τ is the temperature hyper-parameter.

In practical implementation, to improve computational efficiency and save GPU memory, we no longer set the cache queue of negative samples like the MoCo strategy[11] but perform operations on instances within the same mini-batch. From the perspective of matrix operations, it can be expressed as follows:

$$\ell_{i,j}^m = -\log \frac{\exp[(\sum_{column} (H_i^m * H_j^m))/\tau]}{\exp[(\sum_{column} (H_i^m \times H_j^{m\top}))/\tau]} \quad (11)$$

where $H_i^m \in \mathbb{R}^{n \times d}$ and $H_j^m \in \mathbb{R}^{n \times d}$ represent the matching representations of the nodes in the current mini-batch under channel i and j , respectively. n is the size of the mini-batch, $*$ is Hadamard product, \times is matrix multiplication, $\sum_{column} A$ represents the column-wise summation of the two-dimensional matrix A . τ is the temperature hyper-parameter.

Therefore, the final cross-channel matching self-supervised contrastive loss function under the three channels is:

$$\ell_{s11} = \ell_{s,j}^m + \ell_{s,p}^m + \ell_{p,j}^m \quad (12)$$

To preserve the characteristics of the matching representation under different channels, SSL based on maximum mutual information is performed in each channel. Yu et al.[43] point out in their research that the maximum mutual information model used in DGI[28] is established directly between the target node representation and the channel-wise graph representation. This is a relatively coarse-grained strategy, which cannot guarantee that the encoder distills sufficient information from the inputs. Especially when facing a large-scale graph, there is more significant difference between the node representation and the entire graph, the maximum mutual information will lose its effect, and even play a negative role. Therefore, this paper adopts the HMIM model proposed by Yu et al.[43] to establish a two-level maximum mutual information model in each channel. The first level is the maximum mutual information from the node to its corresponding ego-network, and the second level is from the ego-network to corresponding graph. Therefore, we define the self-supervised loss function in a single channel as:

$$\ell_{s12} = - \sum_{i \in \{s,j,p\}} \{ \sum_{u \in U'} \log \sigma[(h_{u,i}^m * z_{u,i}^m) - (\tilde{h}_{u,i}^m * z_{u,i}^m)] + \sum_{u \in U'} \log \sigma[(z_{u,i}^m * z_i^m) - (\tilde{z}_{u,i}^m * z_i^m)] \} \quad (13)$$

where $U' \subset U$ is the user set of the current mini-batch, $h_{u,i}^m \in \mathbb{R}^d$ is the matching representation of user u under channel i , $z_{u,i}^m = \sum_{v \in N_u^i \cup \{u\}} e_{uv}^i * h_v^m$ is the matching representation of the ego-network of user u under channel i , e_{uv}^i indicates the weight of the edges between u and v under channel i . $\tilde{h}_{u,i}^m$ is the negative sample of $h_{u,i}^m$, which is obtained by the row-wise or column-wise shuffling of $h_{u,i}^m$, so is the $\tilde{z}_{u,i}^m$. $z_i^m = \text{AveragePooling}(H_i^m)$ is the average of the matching representation of all nodes in the graph under channel i . We sum the SSL loss of the inner-channel and the cross-channel as the comprehensive SSL loss of the matching representation. And we regard it as the loss of auxiliary task 1:

$$\ell_1 = \ell^m = \ell_{s11} + \ell_{s12} \quad (14)$$

3.3 Hierarchical Attention for Comprehensive User Representation

In this section, we need to fuse two levels of user information, namely, the common representation and matching representation of each user within one channel, and the information of different channels. An intuitive idea is to simply perform summation, average or maximum, but the valuable information for the target task cannot be distilled for each user during the training process. Therefore, this paper proposes a hierarchical attention information fusion model in this section, which learns to extract the most valuable information from multi-levels.

First, we need to fuse the common representation and matching representation of users under each channel. For each user, we learn the attention coefficients (α_s^m, α_s^n) , (α_j^m, α_j^n) , (α_p^m, α_p^n) respectively, which represent the weight of the user's common representation and matching representation under different channels. The attention function in this section can be defined as follows:

$$\alpha_c^q = f_{att1}(h_c^q) = \frac{\exp(a^\top \cdot W_{att1} h_c^q)}{\sum_{q \in \{m,n\}} \exp(a^\top \cdot W_{att1} h_c^q)} \quad (15)$$

where $a \in \mathbb{R}^d$ and $W_{att1} \in \mathbb{R}^{d \times d}$ are learnable parameters. Finally, the user representation under each channel is expressed as:

$$h_c = \sum_{q \in \{m,n\}} \alpha_c^q h_c^q \quad (16)$$

Secondly, after obtaining the user representation under each channel, the user representations of different channels need to be fused. Like the user information fusion within one channel, we adopt the multi-channel fusion technology based on the attention mechanism, and finally get the comprehensive user representation of the multi-channel:

$$h = \sum_{c \in \{s,j,p\}} \alpha_c h_c \quad (17)$$

where α_c is the attention coefficient of the corresponding channel, and its learning method is similar to equation (15).

3.4 Overall Optimization

After obtaining the user's comprehensive representation (section 3.3), we use the interactive matrix R between the users and the items to obtain the final item representation:

$$Q^{(l+1)} = D_i^{-1} R^\top H^{(l)} \quad (18)$$

where $D_i \in \mathbb{R}^{n \times n}$ is the degree matrix of R .

Except for the hierarchical self-supervised learning tasks (auxiliary task 1) based on cross-channel matching representing. We refer to the MHCN[43] and use the HMIM model for self-supervised

ALGORITHM 1: The process of CMHC.**Input:** User-Item graph adjacency matrix R and Social Graph adjacency matrix A .**Output:** Recommendation prediction result \hat{R} .

1. Initialize all parameters
2. Obtain the relationship A_s, A_j, A_p between users under according to Table 1
3. **for** $i=1,2,3,\dots,epoch$ **do**
4. Obtain the common representation H_s^n, H_j^n, H_p^n according to equation (2).
5. Obtain the matching representation H_s^m, H_j^m, H_p^m according to equation (4).
6. Obtain ℓ_{s11} , the loss of cross-channel contrastive learning based on matching representation according to equation (13).
7. Obtain ℓ_{s12} , the loss of inner-channel contrastive learning based on matching representation according to equation (14).
8. Obtain $\ell_1 = \ell^m = \ell_{s11} + \ell_{s12}$, the overall loss of contrastive learning based on matching representation (Auxiliary task 1).
9. Obtain $(\alpha_s^m, \alpha_s^n), (\alpha_j^m, \alpha_j^n), (\alpha_p^m, \alpha_p^n)$, the coefficients of the user's normal representation and matching representation under different channels according to equation (16).
10. Obtain h_s, h_j, h_p , the user representation under each channel according to equation (17).
11. Obtain h , the user comprehensive representation over multi-channel according to equation (18).
12. Obtain Q , the final item representation according to equation (19).
13. Obtain ℓ_2 , the overall loss of contrastive learning based on user common representation (Auxiliary task 2).
14. Obtain \hat{R} , the recommendation prediction result.
15. Obtain ℓ^r , the loss of recommendation prediction (Main task).
16. Obtain L , the loss of joint learning.
17. Use Adam optimizer and backpropagation algorithm to optimize model parameters.
18. **end for**
19. **Return** \hat{R} .

learning, and use it as an auxiliary task 2:

$$\ell_2 = - \sum_{i \in \{s,j,p\}} \left\{ \sum_{u \in U} \log \sigma[(h_{u,i}^n * z_{u,i}^n) - (\tilde{h}_{u,i}^n * z_{u,i}^n)] + \sum_{u \in U} \log \sigma[(z_{u,i}^n * z_i^n) - (\tilde{z}_{u,i}^n * z_i^n)] \right\} \quad (19)$$

where U is the user set of the current mini-batch, $h_{u,i}^n \in \mathbb{R}^d$ is the common representation of user u under channel i , $z_{u,i}^n = \sum_{v \in N_u^i \cup \{u\}} e_{uv}^i * h_{v,i}^n$ is the common representation of the ego-network of user

u under channel i , e_{uv}^i indicates the weight of the edges between u and v under channel i . $\tilde{h}_{u,i}^n$ is the negative sample of $h_{u,i}^n$, which is obtained by the row-wise or column-wise shuffling of $h_{u,i}^n$, so is the $\tilde{z}_{u,i}^n$. $z_i^n = \text{AveragePooling}(H_i^n)$ is the average of the common representation of all nodes in the graph under channel i .

For the main task, use the BPR loss function[27] to optimize the model:

$$\ell_r = \sum_{i \in I(u), j \notin I(u)} -\log \sigma(\hat{r}_{u,i} - \hat{r}_{u,j}) + \lambda \|\theta\|_2^2 \quad (20)$$

where $I(u)$ is the set of items purchased by user u , θ is the model parameters, λ is regularization coefficient. In each training process, a randomly selected positive sample item i purchased by

user u and a undetected item j form a triple for training optimization. $\hat{r}_{u,i}$ is the recommendation prediction between u and i , which is calculated by the representation of user and item:

$$\hat{r}_{u,i} = h_u^\top q_i \quad (21)$$

In summary, the auxiliary tasks 1 and 2 are jointly trained with the main task to optimize the model:

$$L = \ell_r + \beta_1 \ell_1 + \beta_2 \ell_2 \quad (22)$$

where β_1 and β_2 are the coefficients of auxiliary task 1 and auxiliary task 2, respectively. Algorithm 1 illustrates the main steps that are applied to optimize the framework.

4 EXPERIMENTS

In the experiment section, we will conduct various experiments (including performance comparison experiments, ablation experiments, and parameter sensitivity experiments) on real datasets to answer the following questions:

- **RQ1:** How does the method proposed in this paper as compared with the state-of-the-art recommendation methods?
- **RQ2:** What are the benefits of the cross-channel matching representation and hierarchical SSL on cross-channel information proposed in this paper?
- **RQ3:** How do different hyperparameter settings affect the model?

4.1 Experimental Protocol

4.1.1 Datasets. In the experiment, we select three public real datasets commonly used in recommender systems: Douban[45], Epinions[8], and Yelp[43] to verify different methodologies. The task of the experiment is to perform Top-10 recommendations on the processed open-source data that comes from[27, 43]. The statistical information of the experimental data is shown in Table 2.

Table 2. The statistical information of the experimental data.

Dataset	#Users	#Items	#Ratings	#Relation	Density
Douban	2,848	39,586	894,887	35,770	0.794%
Epinions	1,508	2,071	35,500	1,854	1.137%
Yelp	19,539	21,266	450,884	864,157	0.11%

4.1.2 Baselines. To fully verify the performance of the CMHC framework proposed in this paper, we compare it with the most advanced and commonly used baseline methods. There are MF-based methods, GNN-based methods, methods with SSL auxiliary tasks, and methods without SSL auxiliary tasks. It is worth noting that SGL[35] and S^2 -MHCN[43] are SOTA methods. It should be noted that the following baseline methods are from the open-source library QRec⁴.

- BPR[27]. This is a classic recommendation method. It first proposed a general optimization method based on Bayesian Personalized Ranking. This paradigm has been used by subsequent research and is the basis of the ranking recommendation algorithms.
- SBPR[46]. This is an optimized version based on the BPR model[27]. It integrates social relationships into the BPR model and tends to give higher rankings to items liked by the friends of the target user. Compared with BPR, it achieves better performance in both general and cold-start scenarios.

⁴<https://github.com/Coder-Yu/QRec>

- NGCF[32]. This is a collaborative filtering recommendation algorithm based on GNN. It believes that the high-order relationship of user and items contains rich collaborative filtering information. Its core idea is to incorporate the high-order relationship into the process of representation learning, to better integrate collaborative filtering information into users and items.
- DiffNet[36]. This is also a representative model of integrating social relationships into recommendations. However, it believes that the social influence of the user is dynamic, and this influence makes the interest of involving users constantly changing in the process of diffusion. Based on this, a social influence communication model is designed to improve the performance of the recommendation model.
- LightGCN[12]. This is a recent novel study. In this work, the author studied GNN-based collaborative filtering models such as NGCF[32] and found that the two standard operations(model-feature conversion and nonlinearity operations) in the GNN are not necessary for collaborative filtering, but increase the difficulty of training. Therefore, by simplifying the GNN model, they improved the recommendation performance of collaborative filtering while reducing training difficulty and computational complexity.
- SGL[35]. It is the latest exploration to use SSL to solve the limitations of GNN-based recommendations under the supervised learning paradigm. It believes that existing GNN-based recommendation methods for learning the representation of use-item graphs have limitations in solving long-tail recommendations and resisting noise interaction. It designs three strategies of Node-Drop, Edge-Drop, and Random-Walk to achieve data augmentation, and obtains different graph structure views to construct SSL auxiliary tasks, which can effectively break through the limitations brought by GNNs.
- S²-MHCN[43]. This is the latest social recommendation model with self-supervised learning. It combines social networks with user-item graphs, constructs three-channel information, and encodes each channel through the hypergraph neural network. On this basis, it performs self-supervised learning through its proposed HMIN model in each view, which is regarded as an auxiliary task. But it does not fully explore the relationship between different channels, which is the significant difference with this paper.

4.1.3 Metrics. To comprehensively evaluate the models, we selected three commonly used metrics in the recommender systems, namely: Precision@10, Recall@10, and NDCG@10.

4.1.4 Experiment Settings. For the baseline methods, we use grid search to determine the best parameter settings. The maximum iteration is searched in {10,20,30,40,50,60,70,80,90,100}, for other parameters, we use the best settings suggested in the original paper to maintain its optimal performance in real scenarios⁵. For the general settings of all methods involved in the experiment, the user and item embedding dimensions d are set to 50, the batch size is set to 2000. For our method, the regularization coefficient λ is set to 0.03, the SSL loss coefficient β_1, β_2 are searched in {0.001,0.002,0.003,0.005,0.01,0.02,0.05,0.1}, and finally, the optimal values are determined as $\beta_1=0.01$ and $\beta_2=0.001$. The temperature smoothing coefficient τ is searched in {0.01,0.02,0.05,0.1,0.2,0.5,0.6,0.7,0.8}, and the optimal value is finally determined as $\tau=0.5$. The maximum iteration is determined as 30, and the model layer is set to 2. To ensure the stability, comprehensiveness, and credibility of the results, the result of each experiment is the average result

⁵To make the result more reasonable, we unify the parameters under all datasets instead of setting the corresponding optimal parameters for specific dataset. For example, on FilmTrust dataset, we don't increase the layer of CMHC to improve the metric values, as shown in Figure 10.

Table 3. General recommendation performance result.

Dataset	Metric	BPR	SBPR	NGCF	DiffNet	SGL	LightGCN	S^2 -MHCN	CMHC	Improv.
Douban	P@10	14.45%	18.50%	18.34%	17.19%	19.45%	20.46%	<u>20.76%</u>	21.18%	+2.02%
	R@10	2.98%	4.30%	4.30%	4.13%	5.00%	5.24%	4.82%	<u>5.10%</u>	-2.67%
	N@10	0.1669	0.2120	0.2069	0.1929	0.2227	0.2322	<u>0.2358</u>	0.2413	+2.33%
FilmTrust	P@10	25.62%	25.24%	24.37%	25.72%	0.21%	<u>26.19%</u>	26.00%	27.33%	+4.35%
	R@10	48.49%	46.03%	43.45%	48.43%	0.85%	<u>49.59%</u>	49.11%	51.34%	+3.53%
	N@10	0.5687	0.5208	0.4778	0.5618	0.0052	<u>0.5805</u>	0.5797	0.5867	+1.07%
Yelp	P@10	1.20%	1.72%	2.19%	2.00%	2.67%	2.41%	<u>2.87%</u>	2.90%	+1.05%
	R@10	3.41%	4.19%	5.57%	5.27%	7.05%	6.36%	<u>7.46%</u>	7.57%	+1.07%
	N@10	0.0246	0.0339	0.0433	0.0403	0.0566	0.0496	<u>0.0604</u>	0.0609	+0.83%

Table 4. Cold-start recommendation performance result.

Dataset	Metric	BPR	SBPR	NGCF	DiffNet	SGL	S^2 -MHCN	LightGCN	CMHC	Improv.
Douban	P@10	0.83%	1.46%	1.30%	1.33%	1.69%	1.60%	<u>1.70%</u>	1.74%	+2.35%
	R@10	3.62%	6.56%	5.72%	5.73%	7.31%	7.35%	<u>7.54%</u>	7.74%	+2.65%
	N@10	0.0252	0.0468	0.0407	0.0417	0.0504	0.0521	<u>0.0541</u>	0.0555	+2.59%
FilmTrust	P@10	10.51%	9.38%	8.70%	10.34%	0.13%	10.54%	<u>10.74%</u>	11.05%	+2.89%
	R@10	44.42%	39.35%	36.79%	43.66%	1.02%	44.70%	<u>45.50%</u>	46.93%	+3.14%
	N@10	0.4612	0.4012	0.3554	0.4502	0.0053	<u>0.4733</u>	0.4704	0.4906	+3.66%
Yelp	P@10	0.78%	0.98%	1.27%	1.21%	1.65%	<u>1.73%</u>	1.48%	1.75%	+1.16%
	R@10	3.63%	4.32%	5.71%	5.45%	7.47%	<u>7.87%</u>	6.66%	7.95%	+1.02%
	N@10	0.0.16	0.0279	0.0364	0.0349	0.0506	<u>0.0525</u>	0.0435	0.0535	+1.90%

of 5 cross-validations, and without special instructions, the result is the average of the 10 times experiments.

4.2 Performance Comparison(RQ1)

In this section, we use performance comparison experiments to verify whether the CMHC framework proposed in this paper can outperform the most advanced recommendation methods. As analyzed previously, self-supervised learning can not only boost the recommendation performance in general scenarios by mining richer information but also can alleviate the problem of data sparseness in cold-start scenarios with the help of the correlation between multi-channel information. Therefore, we verify the performance of different methods on the complete datasets in the general scenario and the sparse data in the cold-start scenario. The sparse data in the cold-start scenario only retain users with less than 20 interactions. The experimental results are shown in Table 3 and Table 4, respectively, which is the best performance of 10 times experiments. Bold represents the best result; underline represents the sub-optimal result. By analyzing the results, we can draw the following conclusions:

- The CMHC framework proposed in this paper achieves the best performance among all methods. CMHC outperforms S^2 -MHCN (the most competitive baseline) by a distinct margin and the improvement is more obvious in the cold-start scenario, which verifies the performance the benefits of the cross-channel matching representation module and the hierarchical self-supervising module proposed in this paper (This is also verified in section 4.3.2). LightGCN is a very competitive baseline method. Its performance is comparable to that of S^2 -MHCN, and it is a lighter model with lower computational complexity.

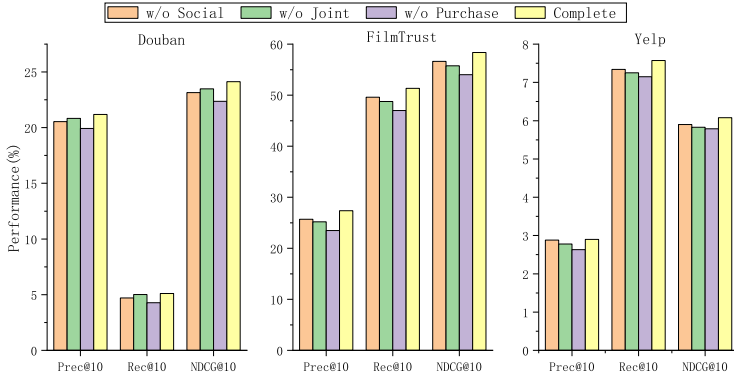


Fig. 6. Schematic diagram of the contribution of different channel information on different datasets to the performance.

- The overall performance of SSL methods, such as SGL, S^2 -MHCN and CMHC, is better than other methods, especially in the cold-start scenario, which strongly proves the benefit of self-supervised learning on the recommendation performance. Various self-supervised learning tasks can autonomously mine the potential characteristics from the raw data or the associated information between different channels, which can alleviate the problem of sparsity. This is particularly important in the recommendation scenario of massive data, and it also shows that research on self-supervised learning is a promising work in the field of recommender systems.
- The GNN-based methods perform better than BPR and SBPR. Graph data has more advantages than traditional European data in relational modeling. The high-order relation mining ability of hypergraph has further improved the recommendation performance. This is also an important reason for the rapid industrial application of GNN-based methods in recommender systems in recent years.
- Methods that incorporate extra information such as social relationships, such as SBPR, DiffNet, S^2 -MHCN, and CMHC possess more satisfactory results. At present, social networks have become the most effective carrier for mining user interests and potential behavior characteristics. The historical behavior of users on social networks can provide the most effective reference for future recommendations. This can also explain that although SGL uses self-supervised learning to promote the recommendation task, its performance does not meet expectations, especially on the FilmTrust data set.

4.3 Ablation Study (RQ2)

First, one of the highlights of this paper is to make full use of multi-channel information for self-supervised learning, to maintain the unique characteristics of different channels, and mine the associated information between different channels. Therefore, we first conducted an ablation study on multi-channel information. Specifically, we get different variants by removing different channels. For example, w/o Social means removing the Social-channel and retaining the Joint and Purchase channels. By comparing the performance of different variants and the complete model which includes all channels on different datasets, the contribution of different channels to the performance is verified. The experimental results are shown in Figure 6. By analyzing the results in Figure 6, we can draw the following conclusions:

4.3.1 Effects of the Multi-channel Information.

- On all datasets, the complete model consistently outperforms the three variants that remove the single-channel data, which proves that the multiple channels are reasonable and efficient. This conclusion is consistent with MHCN[43].
- By comparing the three variants, we can find that their performances on different datasets are different. For example, on the Douban dataset, w/o Joint achieves the best performance, w/o Social is the second, and w/o Purchase achieves the worst performance. This shows that the value of Purchase-channel information is the largest on the Douban dataset, and the Social-channel is the second, Joint-channel is the smallest. The situation on other datasets is different, which shows that the information of different channels plays different roles on different datasets. This will be further analyzed in the experiment of the attention mechanism (Figure 7).

To further verify the contribution of the information of different channels on different datasets to the performance, the attention coefficients of different channels are recorded during the experiment. We draw the data distribution as shown in Figure 7. It can be seen from Figure 7 that on different datasets, the information of different channels does play different roles. On three datasets, Purchase-channel plays the most important role. On FilmTrust and Yelp, most attention values of Social-channel are close to 0, which is consistent with the conclusion of MHCN[43]. The major difference is that the contribution of Social-channel on the Douban dataset is higher than that of the Joint-channel. We believe that the reason may be that the addition of cross-channel matching representation learning, and hierarchical SSL makes the model mine more important social information. Jointly analyzing Figure 6 and Figure 7, we can find that the attention distributions on the datasets in Figure 7 are consistent with the performance of the corresponding variants in Figure 6. This again verifies the rationality of the multi-channel attention fusion model, that is, it can learn the attention that better integrates the information of each channel and maximizes the performance.

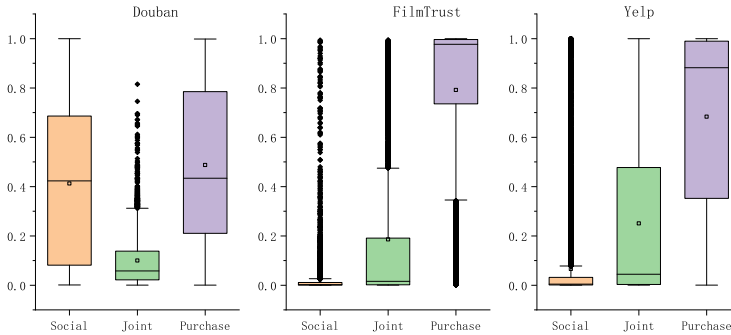


Fig. 7. Attention distribution of different channels on different datasets.

4.3.2 Effects of the Proposed Core Components. To comprehensively verify the rationality and effects of the core components of the model proposed in this paper, we first conduct ablation experiments, and construct variants of the CMHC by removing different core components. $\text{CMHC}_{w/oME\&ME.ssl}$ represents the variant obtained by removing both the matching representation module (section 3.1) and hierarchical SSL (section 3.2) from CMHC. $\text{CMHC}_{w/oME.ssl}$ is the variant obtained by removing the hierarchical SSL model (section 3.2) from CMHC. Compared with $\text{CMHC}_{w/oME\&ME.ssl}$, it adds the matching representation module (section 3.1). CMHC is the complete model proposed in this paper that includes both matching representation module (section 3.1) and hierarchical SSL model (section 3.2). The performance of the above three variants is shown in Figure 8. By analyzing the results in Figure 8, we can get the following conclusions:

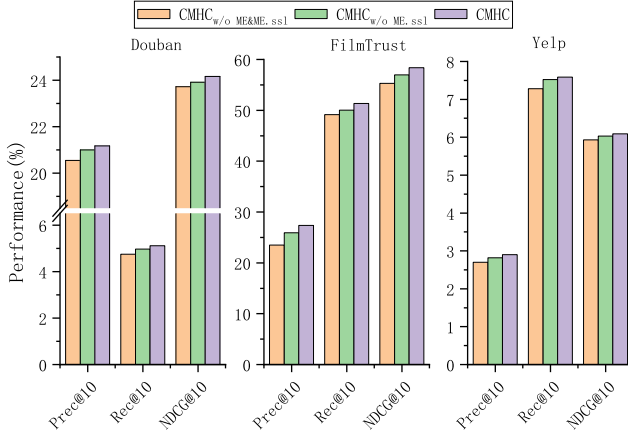


Fig. 8. The performance of different variants on different datasets.

- Compared with $\text{CMHC}_{w/o \text{ME}\&\text{ME}_{ssl}}$, $\text{CMHC}_{w/o \text{ME}_{ssl}}$ brings consistent and significant performance improvements on all metrics and all datasets. This shows that the matching representation module (section 3.1) can dig out the associated information between different channels by learning the cross-channel matching representations, which is extremely valuable for recommendation tasks, even if it has not been leveraged by self-supervised tasks (section 3.2).
- Compared with $\text{CMHC}_{w/o \text{ME}_{ssl}}$, the performance on all datasets of CMHC has been further improved. This shows that the hierarchical SSL model (section 3.2) proposed in this paper can make full use of the feature information captured by the matching representation module (section 3.1) to establish an effective self-supervised task, and as an auxiliary task to promote the performance of the main task.
- Through the joint analysis of Figure 6 and Figure 2, we can see that a reasonable design of the matching representation model between different channels and the self-supervision task based on this can not only eliminate the performance decline problem caused by directly performing contrastive learning on different channel and can bring significant performance improvements. This shows that the original purpose of this paper has been achieved, which again verifies the exploration and analysis conclusions of this paper in Preliminaries 2.3.

4.4 Parameter Sensitivity Analysis (RQ3)

4.4.1 Coefficients of SSL. In this section, we first explore the two most important hyperparameters: β_1 , the coefficients of hierarchical self-supervised loss based on cross-channel matching representations (auxiliary task 1) and β_2 , the coefficients of self-supervised loss based on intra-channel common representations (auxiliary task 2). Figure 9 clearly depicts the performance of different combinations of β_1 and β_2 on different datasets. On the three datasets, the overall performance of the model is relatively stable, and it has a good tolerance for the selection of β_1 and β_2 . The adjustment of the two parameters did not cause significant variation in performance. It is worth noting that the two metrics Precision@10 and Recall@10 have the same performance trends as NDCG@10. Due to space limitations, we will not list specific data.

4.4.2 The depth of CMHC. Further, in this section, we investigate the impact of the depth of CMHC. Specifically, we increase the depth from 1 to 5 and draw the performance curves of each metric on different datasets, as shown in Figure 10. From Figure 10, we can see that the model achieves the best performance when the depth is set to 2. As the model deepens, the performance on the

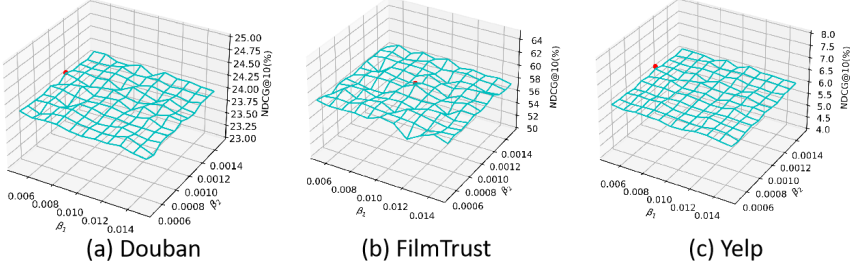


Fig. 9. The performance of CMHC with different combinations of β_1 and β_2 on different datasets.

FilmTrust dataset is slightly higher but the performance on the Douban and Yelp dataset declines obviously, which is consistent with MHCN[43]. The reason is that our proposed framework based on motifs naturally extracts the high-order neighborhood of nodes. Compared with the ordinary GNN model, as the model deepens, the over-fitting phenomenon will appear earlier. This is also in line with the conclusions of exploration on the depth of model in some previous studies[17, 49]. How to solve the over-fitting caused by deep models is also a problem that needs to be further studied in future work about GNNs, especially the Hyper-GNNs.

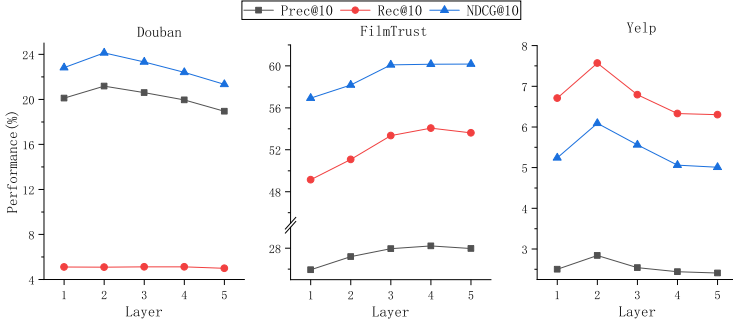


Fig. 10. The performance of CMHC with different depths on different datasets.

4.4.3 Temperature Hyper-parameter of InfoNCE. Finally, we also investigate the impact of different τ values on the performance of CMHC. It can be seen from Figure 11 that when the τ is set too low (below 0.05), all metrics on each dataset will drop dramatically. When the value of τ falls in the range of [0.05, 0.8], the performance of the model is better and more stable, indicating that this range is a reasonable setting. This conclusion is consistent with SGL[35].

5 RELATED WORK

In this section, we will review and analyze the research work related to this paper, including the following aspects:

5.1 Hypergraph and Hypergraph Neural Networks

In recent years, the mining and analysis of graph data has become a research hotspot in the field of data mining and artificial intelligence. Graphs can well represent node attributes and topological structure, which can better model the affinities between nodes[49]. In particular, the emergence of GNNs has accelerated the industrialization process based on graph data research, such as: knowledge question answering and dialogue systems[14, 30], recommender systems [2, 15, 26, 31, 47], intelligent search[4, 20], and etc.

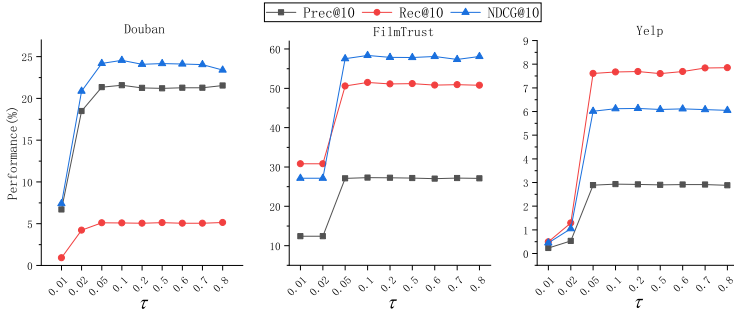


Fig. 11. The performance of CMHC with different τ values on different datasets.

However, with the generation of more and more interactive data, a relationship may not only be limited to two nodes, that is, node pairs can no longer represent some more complex interactive relationships, such as the stable triangular relationship in social networks[42], the joint purchase relationship in the e-commerce networks[43], etc. The hypergraph can break this limitation of the traditional graph, and its edge can connect any number of nodes, and no longer limited to node pairs. Recently, some scholars have also begun to explore modeling methods on hypergraphs. Feng et al.[7] propose a hypergraph neural network representation learning framework, called HGNN, which takes full advantage of the hypergraph fusion of multi-channel data by combining adjacent matrices. Compared with the traditional GNNs, it can better model the multi-channel interactive data of the social network. Some GNN-based methods for recommender systems are proposed recently. Wang et al.[29] pointed out that users may interact with different numbers of items in recommendation scenario, but the traditional graph structure can only represent the pairwise relationship, which is not suitable for a real recommendation scenario. Therefore, they use sequential hypergraphs to model the user's behavior in different periods and use the Hyper-GNN to capture the interaction of multi-hop connections. In addition, through the combination with residual gating and fusion layer, the model can distill user preference more accurately, and the ability of sequence recommendation is significantly improved. Ji et al.[13] argue that whether traditional collaborative filtering recommendation based on matrix factorization or graph-based collaborative filtering recommendation, there are deficiencies in the flexibility of modeling the relationship between users and items and high-order relationships. Based on this analysis, they propose a hypergraph-based collaborative filtering recommendation model, DHCF. The model adopts a dual-channel strategy and uses Jump Hyper-GNN (JHConv) to model users and items. The experimental results prove the value of high-order information for representation learning and the effectiveness of the proposed dual-channel hypergraph model. This paper also uses Hyper-GNNs under different channels to model the high-order interaction between users and items, but it is fundamentally different from DHCF. This paper devotes to model the interaction information and from multi-channel with different patterns, not just interaction information within each channel. More importantly, the focus of this paper is to perform cross-channel matching representation learning to discover the potential association relationships between different interaction patterns, and then make full use of the it for self-supervised learning.

5.2 Self-Supervised Learning

After many years of development, machine learning methods, especially deep learning methods, have achieved great success in image processing[19], natural language processing[18] and other fields. The advantage of deep learning is to mine the valuable features of data from massive amounts of samples, but it requires a large amount of data and labels as input, which makes it difficult to

apply deep learning models to some scenarios where data or label is scarce. The emergence of self-supervised learning (SSL) can well alleviate this problem. The core goal of SSL is to be able to extract valuable information from the data itself or the association between data. When label data cannot be obtained, SSL can play the role of unsupervised learning; in scenarios where limited label data is available, SSL can play a role of pre-training or tuning[39]. SSL was first applied to the field of image processing, such as image restoration and image denoising[38]. SSL methods can be divided into contrastive model and prediction model[21]. The contrastive model usually uses the encoder to encode the data pairs, and then distinguishes the positive instances by calculating the encoding feature distance between the positive and negative pairs (such as the maximum mutual information). The input of the prediction model is generally a single instance. First, a certain method is used to construct a label (generally called as self-generated label), then the data is encoded and predicted, and finally the loss between the predicted label and the generated label is calculated.

At present, with the continuous in-depth research on graph data, some self-supervised learning methods for graph data have been proposed. Shi et al.[33] propose a contrastive learning framework, which encodes node representations from the network view and the meta-path view, respectively, which is used to establish a contrastive learning task. The framework improves the ability to extract both local and high-order structures of nodes. Qiu et al.[25] believe that the attributes of nodes are not transferable, but the structure can be transferred between different graphs. Therefore, they focus on studying how to mine the structural similarity of graphs through SSL tasks and transfer information between different graphs. The GCC framework they proposed is to sample the same node in the same ego network to obtain a positive subgraph, use noise interference from other ego networks to obtain multiple negative subgraphs, and then different GNN-based encoders are used to these subgraphs, based on which the contrastive learning task is established to pre-training the encoders. The experimental results show that the GCC pre-training framework can greatly improve the performance of the initial model.

5.3 Self-supervised Graph Learning for Recommendation

At present, the graph-based recommendation model has become the most popular topic, which greatly boosts the performance of the recommendation. However, this performance improvement is guaranteed by enormous interactive information. On the one hand, with the higher real-time requirements of the recommender systems, the obstacle to obtain the latest high-quality label data in time makes it difficult for the recommender systems to iteratively train in real-time[22]. On the other hand, in large-scale application platforms, new users, or new items have made the cold-start problem more serious[24]. SSL can realize label-free learning with the help of self-supervised tasks, it can also improve the utilization of data, autonomously dig out potentially valuable information, and alleviate real-time recommendation scenarios and cold-start problems to some extent.

The combination of SSL and recommender systems is a new research hotspot in the past two years. Data augmentation is the core of SSL. Due to the continuity of node attributes and the complex relationship between nodes, data augmentation methods in the field of CV and NLP are difficult to directly apply to the recommendation field[35]. The SGL model[35] uses Edge Dropout, Node-Dropout, and Random Walk to generate structural variants of nodes from the original graph. This data augmentation strategy is conducive to capturing the structural pattern of the graph. However, the patterns of social relations between users in the recommender systems can be diversified, so is items. It is difficult to mine the semantic interaction information under various channels in the recommender systems by random strategy.

To explore the high-order interactions and semantic interaction patterns in the recommender systems, some recent studies have begun to combine Hyper-GNNs with SSL. Xin et al.[37] make the first attempt. First, they leverage the hypergraph to construct the session association relationship

between items and then construct the association graph between different sessions, the proposed two-channel Hyper-GNN is used to obtain the feature representations of the two channels, based on which the SSL task is established. The experimental results show that the proposed SSL can significantly improve the recommendation performance as an auxiliary task. Yu et al. propose the MHCN model[43], which uses a hypergraph to encode the features of three channels: Social Motifs, Joint Motifs, and Purchase Motif. Finally, in each channel, the proposed Hierarchical mutual information maximization model is used as a self-supervised auxiliary task to optimize the recommendation model, which greatly improves the recommendation performance.

6 CONCLUSION AND FUTURE WORK

The key issue studied in this paper is how to use multi-channel data to perform more efficient self-supervised learning tasks to enhance the performance of recommendation. First, we make the assumption that constructing a contrastive learning task directly on the features of different channels will make the data of each channel homogeneous, which will deteriorate the recommendation performance. In the exploration experiment, we verified this assumption. To tackle this problem, we propose the CMHC framework, which can make full use of the information in each channel and the information across channels to construct self-supervised learning tasks, thereby improving recommendation performance. Specifically, to comprehensively mine the associated information between different channels while avoiding the problem of homogenization, we propose a cross-channel matching representation learning, which learn the cross-channel matching representation by Attentive-Matching. On this basis, we innovatively proposed a cross-channel hierarchical SSL model based on matching representation, which realized two levels of self-supervised learning within and between channels and improved the ability of self-supervised tasks to autonomously mine different levels of potential information. Finally, we unify the recommendation task (main task), hierarchical self-supervised learning task based on cross-channel matching representations (auxiliary task 1), and self-supervised learning task based on intra-channel common representations (auxiliary task 2) for joint learning. A large number of experimental results on three real datasets show that CMHC outperforms the state-of-the-art methods by big margins, and the ablation studies also prove the benefits of each core components proposed in this paper.

However, in the model depth exploration experiment (section 4.4.2), we also find that as the model deepen, the performance on some datasets drops obviously. The reason is as analyzed in section 4.4.2: our proposed framework based on motifs naturally extracts the high-order neighborhood of nodes. Compared with the ordinary GNN model, as the model deepens, the over-fitting phenomenon will appear earlier. This is also the pain point of hypergraph neural network[17, 49] that need to be addressed. Therefore, how to solve over-fitting caused by the deep model is also a problem that needs to be further studied in future work about graph neural networks, especially the hypergraph neural networks. On the other hand, how to integrate valuable information such as text, sound, images, etc., is also a promising research work in recommender systems.

REFERENCES

- [1] Austin R Benson, David F Gleich, and Jure Leskovec. 2016. Higher-order organization of complex networks. *Science* 353, 6295 (2016), 163–166.
- [2] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The world wide web conference*. 151–161.
- [3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.
- [4] Nurendra Choudhary, Nikhil Rao, Sumeet Katariya, Karthik Subbian, and Chandan K Reddy. 2021. Self-Supervised Hyperboloid Representations from Logical Queries over Knowledge Graphs. In *Proceedings of the Web Conference 2021*. 1373–1384.

- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.
- [6] Xingping Dong and Jianbing Shen. 2018. Triplet loss in siamese network for object tracking. In *Proceedings of the European conference on computer vision (ECCV)*. 459–474.
- [7] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3558–3565.
- [8] G. Guo, J. Zhang, and N. Yorke-Smith. 2013. A Novel Bayesian Similarity Measure for Recommender Systems. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*. 2619–2625.
- [9] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings*, 297–304.
- [10] Harish Haresamudram, Irfan Essa, and Thomas Plötz. 2021. Contrastive Predictive Coding for Human Activity Recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (2021), 1–26.
- [11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9729–9738.
- [12] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. 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*. 639–648.
- [13] Shuyi Ji, Yifan Feng, Rongrong Ji, Xibin Zhao, Wanwan Tang, and Yue Gao. 2020. Dual channel hypergraph collaborative filtering. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020–2029.
- [14] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2021. A Survey on Knowledge Graphs: Representation, Acquisition and Applications. *IEEE Transactions on Neural Networks* (2021), 1–21.
- [15] Jiarui Jin, Jiarui Qin, Yuchen Fang, Kounianhua Du, Weinan Zhang, Yong Yu, Zheng Zhang, and Alexander J Smola. 2020. An efficient neighborhood-based interaction model for recommendation on heterogeneous graph. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 75–84.
- [16] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR (Poster)*.
- [17] Guohao Li, Matthias Müller, Guocheng Qian, Itzel C. Delgadillo, Abdullellah Abualshour, Ali Kassem Thabet, and Bernard Ghanem. 2021. DeepGCNs: Making GCNs Go as Deep as CNNs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), 1–1.
- [18] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering* (2020).
- [19] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. 2021. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems* (2021).
- [20] Jiaying Liu, Jing Ren, Wenqing Zheng, Lianhua Chi, Ivan Lee, and Feng Xia. 2020. Web of scholars: A scholar knowledge graph. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2153–2156.
- [21] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering* (2021).
- [22] Yifei Ma, Balakrishnan Narayanaswamy, Haibin Lin, and Hao Ding. 2020. Temporal-Contextual Recommendation in Real-Time. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2291–2299.
- [23] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. 2002. Network motifs: simple building blocks of complex networks. *Science* 298, 5594 (2002), 824–827.
- [24] Tiejun Qian, Yile Liang, Qing Li, and Hui Xiong. 2020. Attribute graph neural networks for strict cold start recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2020).
- [25] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1150–1160.
- [26] Yanru Qu, Ting Bai, Weinan Zhang, Jianyun Nie, and Jian Tang. 2019. An end-to-end neighborhood-based interaction model for knowledge-enhanced recommendation. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*. 1–9.
- [27] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI '09 Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial*

- Intelligence*. 452–461.
- [28] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. *ICLR (Poster)* 2, 3 (2019), 4.
 - [29] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item recommendation with sequential hypergraphs. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 1101–1110.
 - [30] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.
 - [31] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 950–958.
 - [32] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.
 - [33] Xiao Wang, Nian Liu, Hui Han, and Chuan Shi. 2021. Self-supervised Heterogeneous Graph Neural Network with Co-contrastive Learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.
 - [34] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1437–1445.
 - [35] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised Graph Learning for Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 726–735.
 - [36] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 235–244.
 - [37] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2020. Self-supervised hypergraph convolutional networks for session-based recommendation. *arXiv preprint arXiv:2012.06852* (2020).
 - [38] Yaochen Xie, Zhengyang Wang, and Shuiwang Ji. 2020. Noise2Same: Optimizing A Self-Supervised Bound for Image Denoising. In *Advances in Neural Information Processing Systems*, Vol. 33. 20320–20330.
 - [39] Yaochen Xie, Zhao Xu, Jingtun Zhang, Zhengyang Wang, and Shuiwang Ji. 2021. Self-supervised learning of graph neural networks: A unified review. *arXiv preprint arXiv:2102.10757* (2021).
 - [40] Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. 2019. Cross-lingual Knowledge Graph Alignment via Graph Matching Neural Network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 3156–3161.
 - [41] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems* 33 (2020), 5812–5823.
 - [42] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. 2021. Socially-Aware Self-Supervised Tri-Training for Recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.
 - [43] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021. Self-Supervised Multi-Channel Hypergraph Convolutional Network for Social Recommendation. In *Proceedings of the Web Conference 2021*. 413–424.
 - [44] Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2020. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering* (2020).
 - [45] Guoshuai Zhao, Xueming Qian, and Xing Xie. 2016. User-service rating prediction by exploring social users' rating behaviors. *IEEE transactions on multimedia* 18, 3 (2016), 496–506.
 - [46] Tong Zhao, Julian McAuley, and Irwin King. 2014. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. 261–270.
 - [47] Yujia Zhou, Zhicheng Dou, Bingzheng Wei, Ruobing Xie, and Ji-Rong Wen. 2021. Group based Personalized Search by Integrating Search Behaviour and Friend Network. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 92–101.
 - [48] Dongjie Zhu, Yundong Sun, Haiwen Du, Ning Cao, Thar Baker, and Gautam Srivastava. 2020. HUNA: A method of hierarchical unsupervised network alignment for IoT. *IEEE Internet of Things Journal* 8, 5 (2020), 3201–3210.
 - [49] Dongjie Zhu, Yundong Sun, Haiwen Du, and Zhaoshuo Tian. 2021. MHNF: Multi-hop Heterogeneous Neighborhood information Fusion graph representation learning. *arXiv preprint arXiv:2106.09289* (2021).