electronics

MDPI

Article

Self-Supervised Hypergraph Learning for Knowledge-Aware Social Recommendation

Munan Li [1,*], Jialong Li [2], Liping Yang [1] and Qi Ding [3]

[1] College of Artificial Intelligence, Dalian Maritime University, Dalian 116026, China; liping.yang@dlmu.edu.cn
[2] Department of Computer Science and Engineering, Waseda University, Tokyo 169-8555, Japan; lijialong@fuji.waseda.jp
[3] Global AD Technology Department, Rakuten, Tokyo 158-0094, Japan; qi.ding@rakuten.com
* Correspondence: limunan@asagi.waseda.jp

Abstract: Social recommendations typically utilize social relationships and past behaviors to predict users' preferences. In real-world scenarios, the connections between users and items often extend beyond simple pairwise relationships. Leveraging hypergraphs to capture high-order relationships provides a novel perspective to social recommendation. However, effectively modeling these high-order relationships is challenging due to limited external knowledge and noisy feedback. To tackle these challenges, we propose a novel framework called self-supervised hypergraph learning for knowledge-aware social recommendation (SHLKR). In SHLKR, we incorporated three main types of connections: behavior, social, and attribute context relationships. These dependencies serve as the basis for defining hyperedges in the hypergraphs. A dual-channel hypergraph structure is created based on these relationships. Then, the hypergraph convolution is applied to model the high-order interactions between users and items. Additionally, we adopted a self-supervised learning task to maximize the consistency between different views. It helps to mitigate the model's sensitivity to noisy feedback. We evaluated the performance of SHLKR through extensive experiments on publicly available datasets. The results demonstrate that leveraging hypergraphs for modeling can better capture the complexity and diversity of user preferences and interactions.

Keywords: knowledge-aware social recommendation; dual channel; hypergraph learning; self-supervised learning

Citation: Li, M.; Li, J.; Yang, L.; Ding, Q. Self-Supervised Hypergraph Learning for Knowledge-Aware Social Recommendation. *Electronics* **2024**, *13*, 1306. https://doi.org/10.3390/electronics13071306

Academic Editor: Praveen Kumar Donta

Received: 6 March 2024
Revised: 29 March 2024
Accepted: 29 March 2024
Published: 31 March 2024

1. Introduction

With the rapid popularity of Amazon, Facebook, and other online platforms, recommendation systems [1] have become an essential service in daily life. However, challenges such as the cold start problem and data sparsity have consistently degraded the recommendation performance. As typically used collaborative-filtering (CF)-based methods [2,3] cannot generate effective predictions without sufficient historical interactions, social recommendation algorithms [4,5] integrate users' relationships into representation learning to model users' preference for items. This integration of social relationships enhances the capability of recommendation systems to generate more accurate and personalized suggestions for users.

Recently, the graph neural network (GNN) [6] has shown great power in graph representation learning. The utilization of GNNs facilitates the exploration and learning of the complex interaction patterns between the user's social graph and the user–item rating graph. Consequently, integrating GNN models into the user–item representation learning framework has emerged as a natural approach to enhance social recommendation models [7,8]. However, in real-world applications, data structures often extend beyond simple pairwise connections. This introduces increased complexity in processing and modeling complex high-order data dependencies.

The hypergraph, as an extension of a regular graph structure, has the capability to connect any number of nodes through hyperedges [9,10]. This unique characteristic allows hypergraphs to more effectively capture complex relationships and high-order interactions. Hypergraph-based methods provide a natural and intuitive way to model complex high-order relationships between users and items [11,12]. For example, DHCF [13] was the first proposal for a collaborative filtering framework based on hypergraphs. It leverages the skip hypergraph convolution method to efficiently capture complex high-order correlations. The MHCN [14] introduced a multi-channel hypergraph convolutional network to enhance social recommendation. Compared to traditional graph models, hypergraph models have a significant advantage in handling more complex data interactions.

Despite the exploration of hypergraph structures in handling high-order relations, social recommendation approaches based on hypergraph neural networks still face two primary challenges: (1) Limited external knowledge integration: In real-world recommendation applications, incorporating external knowledge about the items is essential for a more comprehensive understanding of user interests [15,16]. For instance, some movie rating platforms like Douban provide valuable information such as directors, genres, and other related attributes. Leveraging this attribute similarity information could offer additional dependencies to enhance the modeling of high-order interactions in social recommendations. (2) Noisy interactions: Observed relationships and interactions may contain false positive feedback [17,18]. For example, users might mistakenly connect with irrelevant items or form connections based on temporary interests or random actions, which may not accurately reflect user satisfaction. Utilizing such data may amplify the effects of noisy relationships and interactions, making the learning process more susceptible to the noise introduced by these faulty data.

To address the aforementioned challenges, we propose a framework called **s**elf-supervised **h**ypergraph **l**earning for **k**nowledge-aware social **r**ecommendation (SHLKR), which mainly includes three key components: (1) Local relation-aware encoder: We employed a local relation-aware encoder that captures the structure of the local neighborhood and incorporates it into relation-aware embeddings using a graph convolutional network. (2) Global hypergraph relation encoder: We converted the dependencies between users and items into a hypergraph, encoding the possible high-order relationships with multiple types of hyperedges representing different relationship patterns. Next, we employed hypergraph convolution operators to model the high-order relationships within the hypergraph, enriching the latent representations of users and items. To tackle the issue of coupled intents in user's dynamic interests, we introduced a disentanglement layer that aims to separate the intents, ensuring that the learned features do not contain irrelevant noisy information. (3) Cross-view supervision optimization: We designed a cross-view supervision task as an auxiliary task for the recommendation task, utilizing a multi-task training strategy to jointly optimize the model's parameter updates. The task generates contrastive views based on explicit interactions and implicit high-order hypergraphs using an edge dropout operator. By maximizing the mutual information between nodes in different views, it enhances the learning patterns of relationships. These auxiliary supervisory signals can improve the robustness of the learned node representations to cope with edge noise.

Our main contributions are summarized as follows:

- We propose a novel dual-channel hypergraph-learning architecture for knowledge-aware social recommendation. It effectively utilizes hypergraphs guided by different types of relationships to capture high-order interactions among users and items through hypergraph modeling.
- We incorporated a self-supervised learning task that provides auxiliary supervision signals from both the local and global representation spaces. By maximizing the mutual information between different views, we extracted complementary information from each to enhance feature learning through graph convolution.

- Through extensive experiments on benchmark datasets, our proposed method, SHLKR, exhibits significant superiority over state-of-the-art approaches. The results validate the effectiveness and superiority of our proposed model.

2. Related Work

2.1. Graph-Based Social Recommendation

Recent research has focused on leveraging graph data and applied the graph neural network (GNN) [19] to model user–item interactions and social relations [20–22]. As a representative work, He et al. [23] proposed LightGCN, a model that incorporates a lightweight graph convolution layer to capture user–item interactions. It eliminates feature transformation and nonlinear activation to to enhance performance in recommendation models. Due to the sparsity of rating data, many researchers address this challenge by incorporating users' relations as auxiliary data into recommendation models. Among these, Wu et al. [8] introduced DiffNet, a model designed to simulate the dynamic progression of users' latent features through the incorporation of social diffusion. Considering the diverse impacts of users' social relationships, Fan et al. [24] proposed GraphRec, which incorporates an attention mechanism to adaptively learn the influence of social relations. Similarly, Wu et al. [25] introduced DANSER, adeptly employing a graph attention network to effectively capture both static and dynamic preferences. Due to the inherent sparsity of social networks, Shi et al. [26] proposed HERec, a recommendation model grounded in heterogeneous network embedding. It adeptly integrates diverse embeddings derived from a heterogeneous information network with a matrix factorization approach, leading to a substantial performance enhancement. The main idea in the aforementioned studies focused on using graph neural networks (GNNs) to enable the extraction of more complex relationships between users and items.

2.2. Hypergraph Learning for Recommendation

Due to the capacity of hypergraphs to encompass pairwise relationships and effectively model complex higher order dependencies, recent researchers have adopted the strategy of constructing hypergraph structures to capture higher order relationships between users and items [27–29]. To effectively model beyond pairwise relationships, Ji et al. [13] introduced DHCF, a hypergraph collaborative filtering framework. It employs a dual-channel learning strategy and incorporates a skip hypergraph convolution method to efficiently capture the complex high-order correlations. Yu et al. [14] proposed a multi-channel hypergraph convolutional network called the MHCN. It enhanced social recommendation by leveraging hypergraph convolutional to model high-order user interactions. Instead of manually designing and generating hypergraph structures, Xia et al. [30] introduced an automatic hypergraph-structure-learning method called SHT to capture global collaborative relationships. These aforementioned methods aim to improve relationship learning for recommendation by constructing hypergraph connections, effectively capturing complex higher order interactions between users and items.

2.3. Graph Contrastive Learning

One challenge in graph representation learning is the over-smoothing of learned representations due to the stacking of graph message propagation layers [31]. To tackle this issue, graph contrastive approaches integrate a self-supervised learning task to enhance node discriminability [32–34]. In particular, You et al. [35] introduced GraphCL, a graph contrastive learning framework. This approach utilizes data augmentation and consistency maximization between positive and negative pairs to improve graph data representation. Velickovic et al. [36] introduced a self-supervised graph representation learning model. It maximizes mutual information (MI) between global and local graph embedding to enhance graph representation capabilities. Extending self-supervised tasks to graph recommendation models, Wu et al. [17] proposed an approach that supplements the traditional recommendation task with a self-supervision task. This method first generates diverse node views and maximizes the mutual information across these views to reinforce the node representation learning. In summary, the fundamental concept involves training a graph encoder by maximizing the mutual information across various augmented perspectives.

3. Methods

In this section, we provide a detailed overview of the SHLKR model. It mainly includes three main modules: (1) A local relation-aware encoder captures the structure of the local neighborhood and integrates it into relation-aware embeddings using a graph convolutional network. (2) A global hypergraph relation encoder first converts the dependencies between users and items into a hypergraph with various hyperedge types. Next, it utilizes hypergraph convolution to model the high-order relationships within the hypergraph. (3) Cross-view supervision optimization jointly optimizes the model with a self-supervised learning task. The overall framework is illustrated in Figure 1.

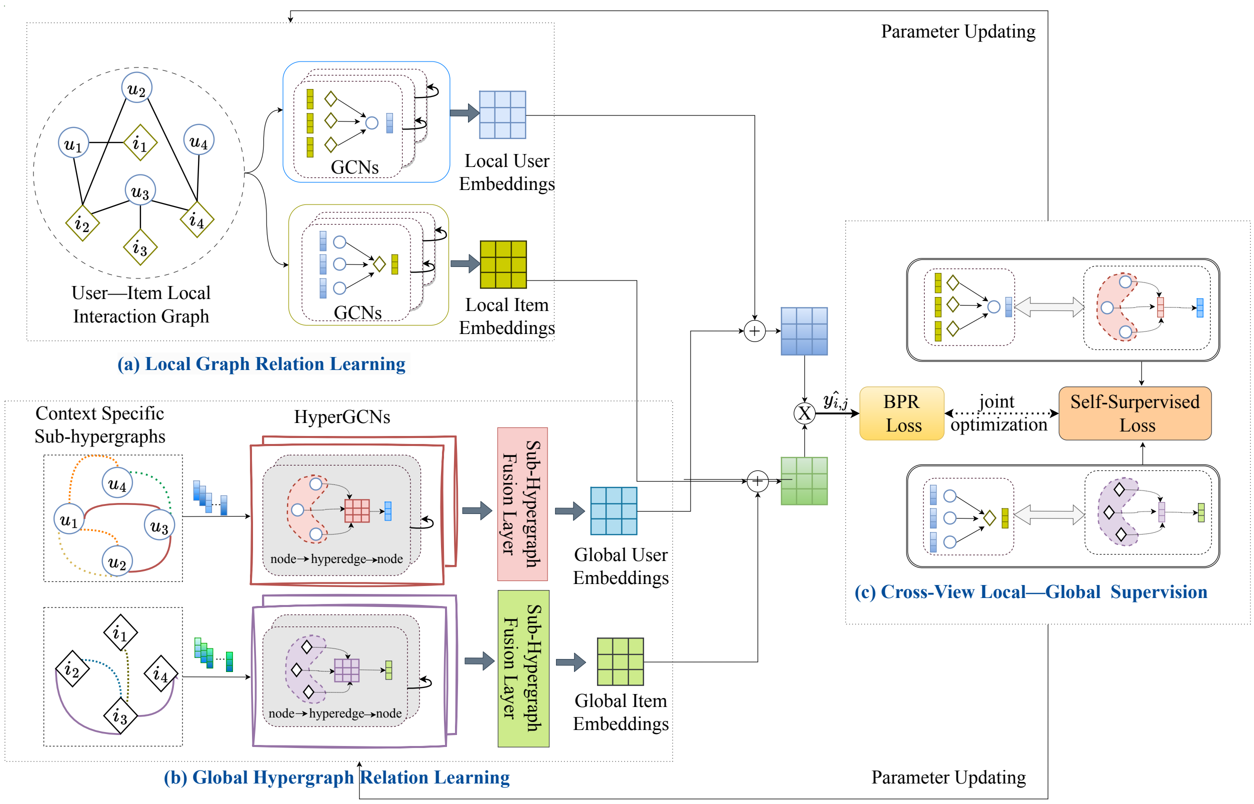

Figure 1. The architecture of the SHLKR model.

3.1. Problem Definition

Let $\mathcal{U} = \{u_1, u_2, \ldots, u_{\mathcal{I}}\}$ denote the user set and $\mathcal{V} = \{v_1, v_2, \ldots, v_{\mathcal{J}}\}$ denote the item set, where $|\mathcal{I}|$ and $|\mathcal{J}|$ are the number of users and items, respectively. Let $\mathcal{G}_r = \{\mathcal{U}, \mathcal{V}, \mathcal{E}_r\}$ denote the user–item interaction graph, where $\mathcal{E}$ is the rating edge set. Additionally, the user's social graph is represented as $\mathcal{G}_s = \{\mathcal{U}, \mathcal{E}_s\}$. $\mathcal{E}_s$ denotes the set of social edges. Similarly, the item's attribute graph is denoted as $\mathcal{G}_a = \{\mathcal{V}, \mathcal{E}_a\}$, where $\mathcal{E}_a$ represents the set of attribute edges. The objective of knowledge-aware social recommendation task is to predict the probability $\hat{y}_{u,v}$ that a given item v will be recommended to the target user u based on $\mathcal{G}_r, \mathcal{G}_s, \mathcal{G}_a$. It can be formulated as

$$\hat{y}_{u,v} = f(u, v, \mathcal{G}_r, \mathcal{G}_s, \mathcal{G}_a; \Theta). \tag{1}$$

Here, Θ is the model learning parameter.

3.2. Local Graph Relation Encoder

According to the prevalent collaborative filtering model, we define $E^u \in \mathbb{R}^{\mathcal{I} \times D}$ and $E^v \in \mathbb{R}^{\mathcal{J} \times D}$ to denote the user and item embedding matrices, respectively. Here, $\mathcal{I}$ and $\mathcal{J}$ denote the numbers of users and items, correspondingly, while D represents the dimensionality of the latent embeddings. We further define a local-graph-relation-encoding layer that captures the local neighborhood structure and incorporates it into the relation-aware embeddings. Let $e^u_{i,l}$ and $e^v_{j,l}$ be the embeddings of user u_i and item v_j at the l-th layer, respectively. We can define the local graph relation encoding layer as follows:

$$
\begin{aligned}
e^u_{i,l} &= \sum_{j \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i||\mathcal{N}_j|}} e^v_{j,l} \\
e^v_{j,l} &= \sum_{i \in \mathcal{N}_j} \frac{1}{\sqrt{|\mathcal{N}_j||\mathcal{N}_i|}} e^u_{i,l},
\end{aligned}
\tag{2}
$$

where $\mathcal{N}_i$ and $\mathcal{N}_j$ are the neighbor sets of user u_i and item v_j, respectively. $\frac{1}{\sqrt{|\mathcal{N}_i||\mathcal{N}_j|}}$ represents the normalization term. Feature transformation and nonlinear activation were eliminated in alignment with LightGCN [23] to streamline the computational costs. Finally, we obtain the user–item local-aware topology embeddings e^u_i / e^v_j as the sum of the user–item embedding at each layer:

$$
e^u_i = \sum_{l=0}^{L} e^u_{i,l}; \quad e^v_j = \sum_{l=0}^{L} e^v_{j,l}. \tag{3}
$$

By stacking multiple local-graph-propagation layers, we enhance user–item representation through the aggregation of local neighborhood information. This process generates contextual embeddings aimed at alleviating the over-smoothing issue inherent in graph neural networks.

3.3. Global Hypergraph Relation Encoder

Motivated by the inherent capability of hypergraphs to capture relationships beyond pairwise connections, we integrated dual-channel hypergraph encoders to effectively model high-order interactions during hypergraph message propagation and aggregation.

3.3.1. Dual-Channel Hypergraphs' Construction

Given user–item rating graph $\mathcal{G}_r$, users' social graph $\mathcal{G}_s$, and items' attribute graph $\mathcal{G}_a$, we initially constructed two channels of homogeneous hypergraphs for users and items separately. Our research is primarily based on three main types of interactions: behavior, social, and attribute context relationships. Based on these three primary types, we can generate different hyperedge structures. By leveraging these hyperedges, we can comprehensively model and capture the complex high-order relationships between users and items.

Figure 2 showcases examples of hyperedge construction in the Douban dataset. Assume, in the user channel, for each pre-defined association rule r_i in the pre-defined list $\{r_1, \cdots, r_k\}$, we generate a hyperedge set $\mathcal{E}_{r_i}$. The hyperedges in $\mathcal{E}_{r_1}$ encompass all nodes $u \in U$ directly connected to each node v. For example, if item i_2 is rated by three users u_1, u_2, u_3, this corresponds to a hyperedge that connects these three users in a homogeneous sub-hypergraph $G^u_{\mathcal{E}_{r_1}}$ within the behavior context. Similarly, in the social context, we can create a sub-hypergraph $G^u_{\mathcal{E}_{r_2}}$, which specifically captures the connections between each user and his/her immediate 1-hop social neighbors. Thus, the relations constructed within a sub-hypergraph are capable of capturing high-order interactions, rather than just pairwise relationships. Following this, we construct sub-hypergraph dependency matrices $\left\{H^u_{\mathcal{E}_{r_1}}, H^u_{\mathcal{E}_{r_2}}, \cdots, H^u_{\mathcal{E}_{r_k}}\right\}$ capturing high-order correlations by utilizing distinct sets of hyperedges.

Analogically, in the item channel, we apply the same approach to form hyperedges, resulting in the sub-hypergraph adjacency matrices denoted as $\left\{H^v_{\mathcal{E}_{r_1}}, H^v_{\mathcal{E}_{r_2}}, \cdots, H^v_{\mathcal{E}_{r_k}}\right\}$. Each matrix $H^v_{\mathcal{E}_{r_i}}$ represents the dependency relationships among items that are either purchased by the same user or share the same attributes. Additionally, this hypergraph structure is extensible, allowing for the addition of new hyperedges to capture more complex patterns of dependencies and interactions as more relationships arise. As a result, this model can generalize to a broader range of data relationships.

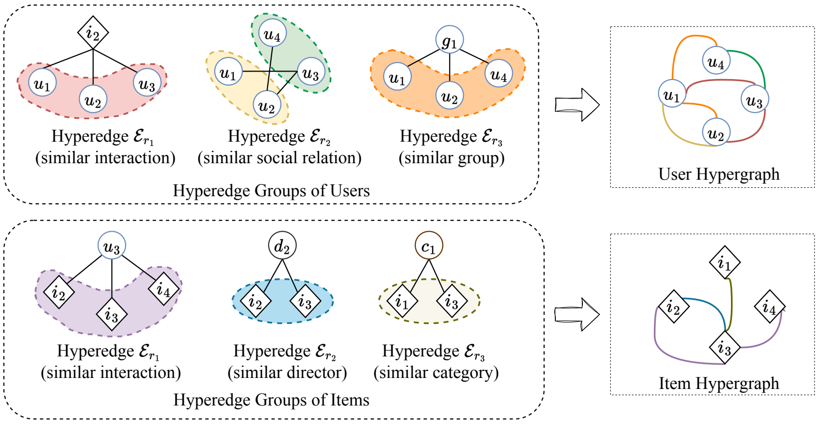

Figure 2. Illustration of diverse hyperedges from both the user and item channel.

3.3.2. Hypergraph Learning in Item Channel

After acquiring sub-hypergraphs with diverse relationship patterns, we introduced a global-aware relation encoder utilizing hypergraph convolution [37]. Formally, the k sub-hypergraph for items is denoted as $H^v_{\mathcal{E}_{r_k}} \in \mathbb{R}^{\mathcal{J} \times \mathcal{E}}$. $\mathcal{H}^v_{\mathcal{E}_{r_k},l} \in \mathbb{R}^{\mathcal{J} \times D}$ represents the hyper-embeddings of items in the k-th sub-hypergraph representation space at the l-th propagation layer. Our hypergraph message propagation and aggregation layer is formulated as follows:

$$\mathcal{H}^v_{\mathcal{E}_{r_k},l} = \mathcal{P}^v_{\mathcal{E}_{r_k}} (E^v_{\mathcal{E}_{r_k}})^{-1} {\mathcal{P}^v_{\mathcal{E}_{r_k}}}^T \mathcal{H}^v_{\mathcal{E}_{r_k},l-1}. \tag{4}$$

Here, $\mathcal{P}^v_{\mathcal{E}_{rk}}$ corresponds to the hypergraph convolution operation, which can be calculated as $\mathcal{P}^v_{\mathcal{E}_{r_k}} = (D^v_{\mathcal{E}_{r_k}})^{-1/2} H^v_{\mathcal{E}_{r_k}}$. $\left\{D^v_{\mathcal{E}_{r_k}}, E^v_{\mathcal{E}_{r_k}}\right\}$ represent the degree matrix for the node and hyperedge, respectively. This allows us to independently learn high-order features of items from each sub-hypergraph. Figure 3 illustrates the hypergraph convolutional layer. The hypergraph convolution involves a two-stage refinement: initially, aggregating node features based on the hyperedge to create the hyperedge feature, implemented through low-rank multiplication with ${H^v_{\mathcal{E}_{r_k}}}^T$; subsequently, aggregating related hyperedge features into node features by multiplying with $H^v_{\mathcal{E}_{r_k}}$. Similar to local convolutional operations, we eliminate feature transformation and nonlinear activation in the hypergraph convolutional layer to reduce the model's complexity.

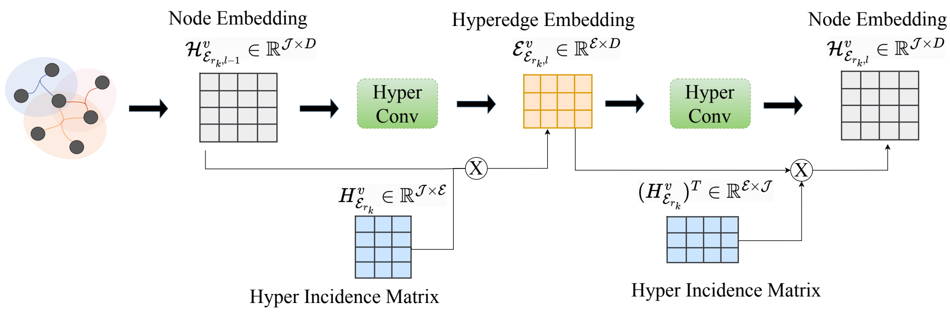

Figure 3. Illustration of the hypergraph convolutional layer in SHLKR.

3.3.3. Hypergraph Learning in User Channel

Analogously, we denote the dependency matrix of the k-th sub-hypergraph for users as $H^u_{\mathcal{E}_{r_k}} \in \mathbb{R}^{\mathcal{I}\times\mathcal{E}}$, where $\mathcal{I}$ represents the number of users and $\mathcal{E}$ represents the number of hyperedges. $\mathcal{H}^u_{\mathcal{E}_{r_k},l} \in \mathbb{R}^{\mathcal{I}\times D}$ represents the hyper-embeddings of users in the k-th sub-hypergraph representation space at the l-th propagation layer. The hypergraph message propagation and aggregation layer in item channel is defined in a way that aligns with the user channel:

$$\mathcal{H}^u_{\mathcal{E}_{r_k},l} = \mathcal{P}^u_{\mathcal{E}_{r_k}}(E^u_{\mathcal{E}_{r_k}})^{-1}\mathcal{P}^{u\ T}_{\mathcal{E}_{r_k}}\mathcal{H}^u_{\mathcal{E}_{r_k},l-1}. \tag{5}$$

Here, $\mathcal{P}^u_{\mathcal{E}_{r_k}} = (D^u_{\mathcal{E}_{r_k}})^{-1/2}H^u_{\mathcal{E}_{r_k}}$ represents the hypergraph convolution operator for users. Directly using the initial embedding as the input for hypergraph learning may contain irrelevant feature information.

To improve the global preference encoder's effectiveness, we further introduce a disentangled preference encoding layer $f_{\varpi}(\cdot)$. It is designed to address the limitations of the global module in capturing multiple intents thoroughly, refining the encoding process for a clearer representation of user preferences and reducing the impact of noisy signals in the interaction data. Suppose that a user's overall preference can be partitioned into m potential preference intents, represented as $q_{u,m} \in \mathbb{R}^d$, where m denotes the specific user preference order. The user's embedding $\mathcal{H}^{u^*}_{\mathcal{E}r_i,l}$ is initially mapped to a specific intent vector through an embedding transformation layer:

$$q_{u,1:m} = \sigma(\mathcal{H}^{u^*}_{\mathcal{E}r_i,l}w_{u,q} + b_{u,q}) \tag{6}$$

where $w_{u,q} \in \mathbb{R}^{d\times M*d}$ and $b_{u,q} \in \mathbb{R}^d$ represent the trainable transformation matrix and bias for user u, respectively. The activation function $\sigma(\cdot)$ is defined as $LeakyReLU()$, specifically as $\sigma(x) = \max(x, \alpha x)$. To calculate the user's attention vector for a specific intent, we utilized a key vector $p_{u,1:m} = \{p_{u,1}, p_{u,2}, \cdots, p_{u,m}\} \in \mathbb{R}^{M\times d}$ to learn the user's attention across feature intents. Here, we calculated the weight through a linear dot-product $q^T_{u,m} \cdot p_{u,m}$. Subsequently, the attention scores are computed using the softmax function:

$$a_{u,m} = \frac{\exp(LeakyReLU(q^T_{u,m} \cdot p_{m,k}))}{\sum_{m=1}^{M}\exp(LeakyReLU(q^T_{u,m} \cdot p_{u,m}))}. \tag{7}$$

Then, we incorporate a memory unit via a memory matrix $z_{u,1:k} = \{z_{u,1}, z_{u,2}, \cdots, z_{u,k}\} \in \mathbb{R}^{M\times d}$. $z_{u,1:k}$, which serves as a storage block. It is used to capture unique preferences across various latent aspects. Through a weighted sum of all disentangled factors, the learned disentangled representations $\mathcal{H}^u_{\mathcal{E}r_i}$ can effectively preserve diverse latent factors associated with different intents in the graph. Specifically, $\mathcal{H}^u_{\mathcal{E}r_i,l}$ is calculated as

$$\mathcal{H}^u_{\mathcal{E}r_i,l} = \sum_{k=1}^{K} a_{u,m} \cdot z_{u,m}. \tag{8}$$

Here, $z_{u,k}$ represents the m-th memory unit of the disentangled factor. It assists the model in capturing diverse user preferences and enhances the learning representation of the global relation encoder.

3.3.4. Sub-Hypergraph Fusion and Prediction Layer

To incorporate the semantics of distinct sub-hypergraphs, we introduce a semantic-level attention layer. This layer aggregates node embeddings from different sub-hypergraphs to generate comprehensive representations. Let $\mathcal{H}_{\mathcal{E}_{r_k},l}$ denote either user or item embeddings for simplicity. The final aggregated representation $\mathcal{H}_l$ is obtained by taking the weighted sum of the embeddings from various sub-hypergraphs. Specially, $\mathcal{H}_{\mathcal{E}_{r_k},l}$ is computed as follows:

$$\alpha_{\mathcal{E}r_k,l} = \frac{\exp(W_a^T \cdot f_\phi(\mathcal{H}_{\mathcal{E}r_k,l}; W_\phi))}{\sum\limits_{\mathcal{E}r_t \in \mathcal{E}r_{|k|}} \exp(W_a^T \cdot f_{W_\phi}(\mathcal{H}_{\mathcal{E}r_t,l}; \phi))},$$
$$\mathcal{H}_l = \sum_{\mathcal{E}r_t \in \mathcal{E}r_{|k|}} \alpha_{\mathcal{E}r_k,l} \cdot \mathcal{H}_{\mathcal{E}r_k,l}. \quad (9)$$

Here, f_ϕ is implemented by a multi-layer perceptron network with learnable weights W_ϕ and a *tanh* activation function. W_a is the learning weight. It is used to determine the importance of each sub-hypergraph's contribution to the final aggregated representation. By aggregating the node features obtained from sub-hypergraphs in the user and item domains separately, we can obtain versatile representations for both users and items. These representations are denoted as ${\mathcal{H}_l}^u$ and ${\mathcal{H}_l}^v$. Then, the embeddings from multiple layers are added together to comprehensively capture features and patterns at different levels. This effectively reveals the complex relationships and semantics in the hypergraph. The final user and item embedding are updated by combining the hypergraph-based representations with the local-aware graph relation embeddings, refined as $P = E^u + \mathcal{H}^u, Q = E^v + \mathcal{H}^v$. Finally, the prediction of user u_i liking item v_j is determined by computing the inner product, $\hat{y}_{i,j} = p_i^T * q_j$.

3.4. Cross-View Local–Global Dependency Supervision

To address the impact of edge noise in graphs, we propose integrating self-supervised learning with the main recommendation task. By maximizing the node mutual information between the generated hypergraph view and the local collaborative view, these auxiliary supervisory signals can improve the robustness of learned node representations. We created self-supervisory signals by considering different views that encode distinct relationship patterns of users. The local collaborative view captures user preference features from explicitly displayed relationship patterns, while the higher order view captures implicit higher order dependency relationships. We have designed an edge dropout operator that randomly drops edges in both local and hypergraph structures with a certain probability, generating different augmented views. Then, we can establish the cross-view contrastive task.

Especially, given a sub-hypergraph embedding $h^u_{\mathcal{E}_{r_k},i}$ and a local-aware topology embedding e_i^u for a specific user i, we employed a shared linear transformation function $\Upsilon(\cdot)$ to map them into a common latent space. This transformation is defined as

$$\delta^u_{k,i} = f_\Upsilon(h^u_{\mathcal{E}_{r_k},i}; \gamma); \eta^u_i = f_\Upsilon(e^u_i; \gamma), \quad (10)$$

where γ is the shared learning parameters. Then, we can formulate the cross-view contrastive loss to maximize the agreement between the representations of positive pairs while minimizing the agreement between the representations of negative pairs. Mathematically, the cross-view contrastive loss for user i can be defined as follows:

$$\mathcal{L}^u_{k,i} = -\ln \frac{\exp\left(s(\delta^u_{k,i}, \eta^u_i)/\tau\right)}{\sum_{i^*=0}^{\mathcal{I}} \exp\left(s(\delta^u_{k,i}, \eta^u_{i^*})/\tau\right)}, \quad (11)$$

where $s(\cdot)$ is used to compute vector similarity, such as the cosine function. The temperature parameter τ adjusts the scaling of the softmax function to increase the discrimination between samples. The positive pairs are defined as nodes that exist in both views, while negative pairs are sampled from different nodes within a batch. For SHLKR, we have k types of guided hypergraphs, the total learning objective for user i can be computed as follows:

$$\mathcal{L}^{\mathcal{U}}_{ssl} = \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \sum_{k=1}^{K} \mathcal{L}^u_{k,i} \quad (12)$$

In the same way, we can also generate the cross-view contrastive loss $\mathcal{L}_{ssl}^{\mathcal{V}}$ for the item channel. The final loss function is defined as the sum of the two losses: $\mathcal{L}_{ssl} = \mathcal{L}_{ssl}^{\mathcal{U}} + \mathcal{L}_{ssl}^{\mathcal{V}}$.

3.5. Model Training

In our training process, we utilized the Bayesian Personalized Ranking (BPR) loss function to optimize the main recommendation task. The BPR loss function is widely used in training ranking models. The definition of the BPR loss function is as follows:

$$\mathcal{L}_{bpr} = \sum_{(i,j^+,j^-)\in D} - \ln \sigma(\hat{y}_{i,j^+}(\Theta) - \hat{y}_{u,j^-}(\Theta)). \tag{13}$$

Here, $\sigma(\cdot)$ is the sigmoid function, Θ includes the whole learning parameters employed in SLHKR, and $D := (u, i, j | i \in I_u^+, j \in I_u^-)$ denotes the set of training samples. To collaboratively guide the parameter updates, we integrated the cross-view self-supervised loss with the BPR loss. The overall objective function is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{bpr} + \lambda_{ssl}\mathcal{L}_{ssl} + \lambda_{\Theta}||\Theta||^2, \tag{14}$$

where λ_{ssl} is used to balance the recommendation task and the cross-view self-supervised task and λ_{Θ} serves as a regularization parameter to prevent overfitting. The learning algorithm for optimizing the proposed model is summarized in Algorithm 1.

Algorithm 1 Learning algorithm of SHLKR.

Require: user–item rating graph, $\mathcal{G}_r$; user's social graph, $\mathcal{G}_s$; item's attribute graph, $\mathcal{G}_a$.
Ensure: user and item embedding matrix, $P \in \mathbb{R}^{\mathcal{I}\times D}$, $Q \in \mathbb{R}^{\mathcal{J}\times D}$; model parameters, Θ.

Hypergraphs' construction, $\left\{H^u_{\mathcal{E}r_k}, H^v_{\mathcal{E}r_k}\right\}$;
Initialize P, Q, Θ;
for *epoch* in *max_epoch* **do**
    Sample *minibatch* training instances;
    for (i, j^+, j^-) in *minibatch* **do**
        for $l = 1, \ldots, L$ **do**
            Update $e^u_{i,l}, e^v_{i,l}$ using Equation (2);
            for each sub-hypergraph **do**
                Update $\mathcal{H}^v_{\mathcal{E}_{r_k},l}$ using Equation (4);
                $\mathcal{H}^u_{\mathcal{E}r_i,l} = f_{\varpi}(\mathcal{H}^{u^*}_{\mathcal{E}r_i,l})$;
                Update $\mathcal{H}^u_{\mathcal{E}_{r_k},l}$ using Equation (5);
            end for
            Update $\mathcal{H}_l{}^u, \mathcal{H}_l{}^v$ using Equation (9);
            $p_{i,l} = \mathcal{H}_l{}^u + e^u_{i,l}, q_{j,l} = \mathcal{H}_l{}^v + e^v_{j,l}$;
        end for
        $p_i = \sum_{l=0}^{L} p_{i,l}$, $q_j = \sum_{l=0}^{L} q_{j,l}$;
        Calculate $\mathcal{L}_{bpr}, \mathcal{L}_{ssl}$;
        $\mathcal{L}_{total} = \mathcal{L}_{bpr} + \lambda_{ssl}\mathcal{L}_{ssl} + \lambda_{\Theta}||\Theta||^2$;
    end for
    Update P and Q and Θ via backward();
end for
return P, Q;

4. Experimental Settings

4.1. Datasets

We evaluated our proposed method using three widely used datasets in social recommendation systems, namely Yelp, CiaoDVD [15], and Douban [38]. Douban is a prominent social media network in China. This dataset consists of multiple types of information, including user ratings for movies, users' social relations, user's group affiliations, and movie attributes such as directors and genres. The Yelp dataset contains user ratings, movie categories, and user social relations. It is derived from a well-known American user review website. The CiaoDVD dataset is collected from a consumer review platform. It contains user ratings for movies and their corresponding categories. Moreover, similar to the previous datasets, the CiaoDVD dataset also provides information about user social relationships. Table 1 provides an overview of the data statistics.

Table 1. Statistics of the datasets.

Dataset	# of Users	# of Items	# of Interactions	# of Relations	Density
Douban	3022	6977	195,493	1366	0.79%
Yelp	14,085	14,037	194,255	150,532	0.46%
CiaoDVD	7375	105,114	216,563	111,781	0.0379%

4.1.1. Baselines

We selected several representative baselines to evaluate the performance of our proposed methods:

- BPR [3]: A ranking model for implicit feedback. It utilizes triplet ranking loss instead of MSE loss, prioritizing target items in the ranking result.
- SBPR [39]: SBPR, an extension of BPR, introduces social relations into the ranking model.
- HERec [26]: A recommendation model based on heterogeneous network embedding that efficiently merges various embeddings derived form a heterogeneous information network with a matrix factorization model.
- DiffNet [8]: A deep influence diffusion model for social recommendation that simulates user's preference diffusion process using a hierarchical-influence-propagation structure.
- LightGCN [23]: A light graph convolution model that eliminates feature transformation and nonlinear activation. It derives the final representation by aggregating multiple layers to mitigate over-smoothing.
- SGL [17]: A self-supervised graph learning model that incorporates auxiliary self-supervised tasks for multi-task learning.
- GraphRec [24]: An attention-based graph model for social recommendation that jointly captures the interactions between the social graph and rating graph, considering the heterogeneous strengths of social relations.
- MHCN [14]: A multi-channel hypergraph convolutional network that enhances social recommendation by leveraging high-order user relations. It integrate self-supervised learning to regain connectivity with hierarchical mutual information maximization.
- DHCF [13]: DHCF introduces a dual-channel learning strategy along with a skip hypergraph convolution method to form a robust hypergraph collaborative filtering framework that efficiently models the complex high-order correlations.

4.1.2. Evaluation Metrics

The experiment employed Recall@K and NDCG@K to assess the algorithm's effectiveness [40]. Recall@K indicates the percentage of accurate predictions present in the Top-K list compared to all predictions. NDCG@K is utilized to assess how well the Top-K recommendations are ranked. Users typically focus on the initial items in the recommended results, aiming for correct predictions to prominently feature in the Top-K prediction list. These two metrics are formulated as follows:

$$\text{Recall@}K = \frac{\text{Number of Hits@}K}{|GT|}, \tag{15}$$

$$\text{NDCG@}K = Z_k \sum_{i=1}^{K} \left(\frac{2^{r_i} - 1}{\log_2(i+1)} \right). \tag{16}$$

4.1.3. Hyper-Parameter Settings

The implemented model utilizes Tensorflow. To ensure a fair experimental comparison, we applied uniform parameter settings across all methodologies. The embedding dimension d was standardized at 64, the regularization coefficient λ fixed at 0.001, and the batch size b maintained at 1500. Additionally, we further explored the model's sensitivity to various hyper-parameters in the experiments. We dynamically adjusted the learning rate to expedite the convergence by employing the Adam optimizer. For graph-based neural network models, the model depth L was consistently set to 2. We employed five-fold cross-validation to obtain the results. The temperature parameter was set at 0.01. The experimental results were obtained using a five-fold cross-validation strategy.

5. Results and Discussion

5.1. Overall Performance Comparisons

In this section, we conduct experimental comparisons on three publicly available datasets to validate the performance of SHLKR and assess its superiority over other benchmark methods. The results are presented in Table 2. The results obtained from the experiments led us to the following observations: (1) SHLKR consistently outperforms other benchmark methods across all three datasets, with improvements of 10.57% and 7.52% on average for the Recall@20 and NDCG@20 metrics, respectively. These results provide strong evidence for the effectiveness of SHLKR. This superiority can be attributed to several key factors. (i) SHLKR transforms the dependency between users and items into a hypergraph representation. It allows SHLKR to encode multiple collaboration pattern and capture more fine-grained interaction modes. (ii) SHLKR parameterizes the global intent of users to separate different intents. It could reduce the impact of noise signals and enhance the representation learning capability of the global relationship encoder. (iii) SHLKR introduces a cross-view supervision task to maximize the mutual information between nodes in different views. It enables the model to learn robustness node embedding against adversarial edge noise. (2) Compared to GNN-based methods such as LightGCN, DiffNet, and GraphRec, SHLKR exhibits superior performance. This is because the hypergraph can leverage hyperedges to connect more nodes, surpassing the limitations of pairwise relationships. In real-world scenarios, the relationships between users and items often involve multiple interaction patterns, including direct interactions, shared interests, social connections, and more. SHLKR is flexible in modeling and representing these complex interaction patterns. (3) Compared to hypergraph methods such as the MHCN and DHCF, SHLKR introduces a greater variety of relationship patterns. In SHLKR, we primarily focus on three main types of interaction: behavioral, social, and attribute context relationships. By incorporating different interaction patterns as additional knowledge, SHLKR is able to more comprehensively capture the complex collaborative signals, thus alleviating sparsity issues. As considering auxiliary interaction relationships may also introduce noise and irrelevant semantics, we additionally propose a disentanglement layer to separate the intents within the learned features, ensuring that irrelevant noisy information is not retained.

Table 2. Overall performance comparison of SHLKR with baselines (Top-k Evaluation).

Dataset	Metrics	Algorithms									
		BPR	**SBPR**	**HERec**	**DiffNet**	**LightGCN**	**SGL**	**GraphRec**	**MHCN**	**DHCF**	**SHLKR (Ours)**
Douban	Recall@20	3.98	4.12	4.28	4.31	4.47	4.62	4.57	4.69	4.79	**5.28** (10.2↑)
	NDCG@20	3.25	3.37	3.41	3.58	3.60	3.76	3.65	3.92	4.02	**4.39** (9.2↑)
	Recall@50	4.43	5.12	5.83	6.20	6.95	7.24	7.58	7.65	7.61	**8.27** (8.1↑)
	NDCG@50	4.55	4.97	5.33	5.87	5.92	6.29	6.37	6.45	6.23	**7.04** (9.4↑)
CiaoDVD	Recall@20	4.02	4.31	4.82	4.45	4.62	4.89	4.97	5.68	5.03	**6.12** (7.8↑)
	NDCG@20	3.07	3.10	3.41	3.29	3.60	3.89	3.96	4.25	3.87	**4.57** (7.5↑)
	Recall@50	5.93	6.14	6.52	7.33	8.61	9.13	8.91	9.32	9.27	**9.69** (6.1↑)
	NDCG@50	3.60	4.09	4.29	4.21	4.78	5.02	5.36	5.47	5.33	**5.82** (6.9↑)
Yelp	Recall@20	5.16	5.41	6.37	7.06	7.24	7.58	7.78	8.28	7.72	**8.83** (6.6↑)
	NDCG@20	3.15	3.68	4.40	4.78	5.03	5.12	5.31	5.47	5.28	**5.79** (5.9↑)
	Recall@50	8.55	9.97	10.33	10.92	11.87	12.67	13.29	13.41	13.37	**14.17** (5.6↑)
	NDCG@50	3.95	4.21	4.82	5.39	5.92	6.23	6.58	6.79	6.61	**7.24** (6.8↑)

↑: Increment of classification accuracy for the corresponding backbone model.

5.2. Model Robustness Analysis

5.2.1. Robustness to Data Sparsity

To further validate the model robustness, we conducted experiments across multiple data splits to ensure the replicability and consistency of the study results across different training data scenarios. Specifically, we partitioned the dataset into subsets with different interaction ratios and trained the model with varying levels of sparsity. The x-axis ticks of Figure 4 provide specific sparsity settings. We then compared SHLKR with the hypergraph-based method DCHF and GNN-based method LightGCN. Additionally, we assessed the performance differences between SHLKR and these two baselines. The results are depicted in Figure 4. We observed that SHLKR's performance is consistently superior to LightGCN and DCHF across datasets with different sparsity levels. This result validated the robustness of SHLKR in handling sparse data in the training data. The reason is that our hypergraph learning model can learn high-order relationships through global relation learning, thereby alleviating the impact of sparsity in the observed user–item interaction data. Additionally, the self-supervised learning task extracts knowledge from refined graph views, providing additional supervised signals to enhance embedding learning.

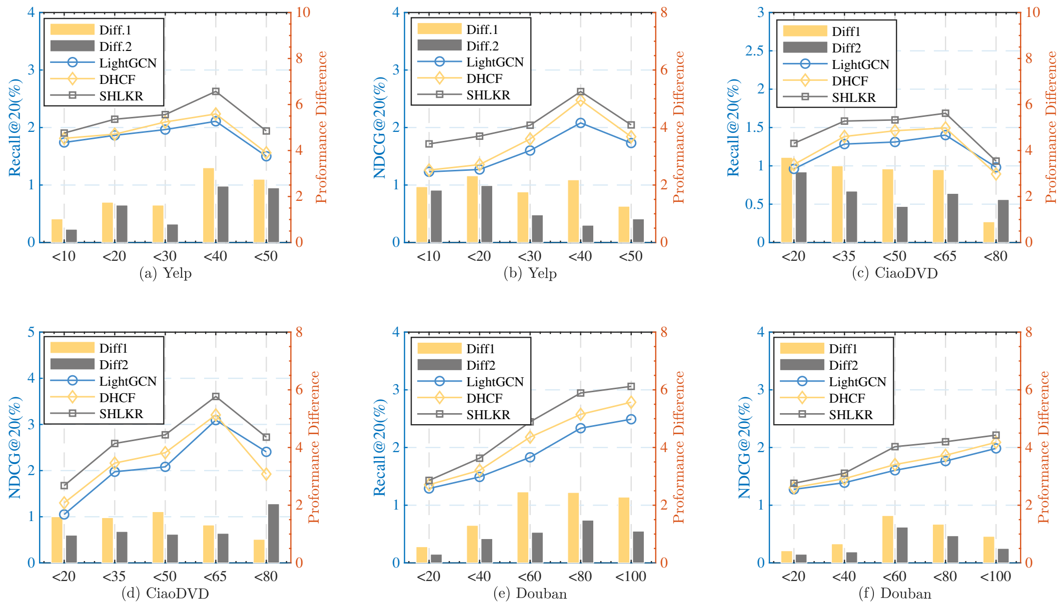

Figure 4. Performance comparison of SHLKR with baselines with respect to different data sparsity levels. The lines represent Recall@20 and NDCG@20, while the bars indicate performance differences between the baselines with SHLKR.

5.2.2. Robustness to Noise Data

In SHLKR, we introduced a self-supervised learning task to enhance node representation learning and improve the model's robustness against noise. To evaluate the effect of the self-supervised learning task on model performance, we introduced different proportions of noise data to the training set. Specifically, the ratio was set to 5%, 10%, 15%, 20%, and 25%. We then compared the performance of our proposed SHLKR model with a variant model without the self-supervised learning task (SHLKR-SSL) using the metric Recall@20. The experimental results are shown in Figure 5. We found that noise data had a negative impact on model performance. However, SHLKR consistently outperformed SHLKR-SSL. This observation indicated that SHLKR consistently demonstrates robust performance at different noise levels. The reason is that the auxiliary task can help SHLKR learn valuable data patterns from cross-view supervision signals. It will enhance the model's stability to noisy edges in graph learning. Figure 6 illustrates the distance distribution of user and item embeddings in two-dimensional space achieved through *t*-SNE visualization. We observed distinct clusters and a clear separation in the embeddings' distribution generated by SHLKR. This further validates the advantage of the self-discrimination task in SHLKR.

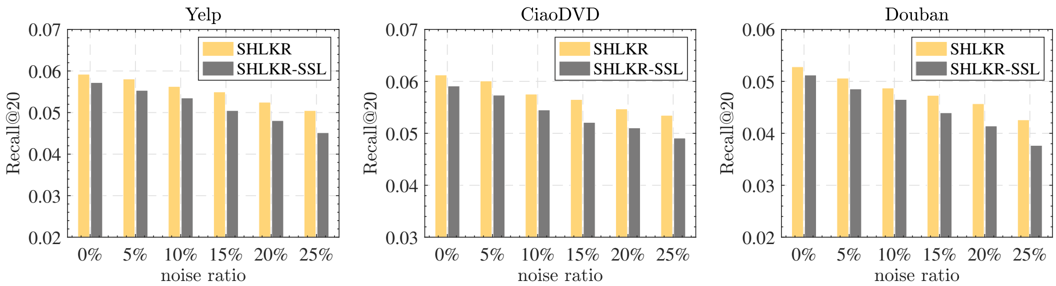

Figure 5. Performance comparison of SHLKR with SHLKR-SSL with respect to the noise ratio.

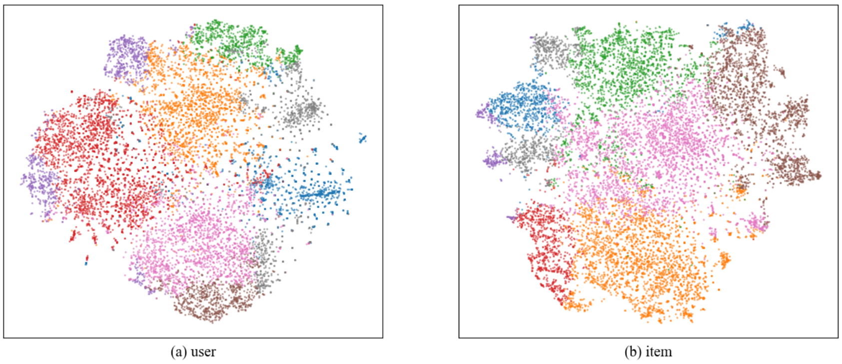

Figure 6. Visualization of embedding using *t*-SNE: (**a**) User and (**b**) Item. Each data point represents a node and is visually distinguished by color. The color reflects its membership in specific clusters identified from Yelp.

5.3. *Hyper-Parameter Sensitivity Analysis*

To gain insight into how the hyper-parameters affect model performance, we conducted experiments on the Yelp dataset to compare results with different hyper-parameter values. This allows us to determine the optimal hyper-parameters to balance the model complexity and performance.

Initially, we changed L from 1 to 4 to assess the influence of graph model depth on SHLKR. The results of Figure 7a,b indicate that the model achieves its best performance when $L = 2$. This result suggest that a relatively shallow graph model is capable of extracting valuable high-order semantics. However, as the model depth increases further, its performance deteriorates. This decline may be due to the over-smoothing caused by the stacking of propagation layers. It will further amplify the impact of edge noise. Figure 7c,d demonstrate that the model achieves its highest performance with a Recall@20 of 0.09251 and a NDCG@20 of 0.0604 when the embedding size is set to $D = 128$. This means setting a small embedding size may prevent the model from capturing sufficient features. As a result, it will affect the generalization capacity of the model to unseen data. Figure 7e,f show the affect of λ_{ssl} on the model performance. The optimal performance is achieved when $\lambda_{ssl} = 0.001$, striking a balance between the main recommendation task and the auxiliary task. When λ_{ssl} is too large, the auxiliary task will have a stronger influence on parameter learning, thus affecting the main task. On the contrary, it may lead to insufficient learning on the auxiliary task.

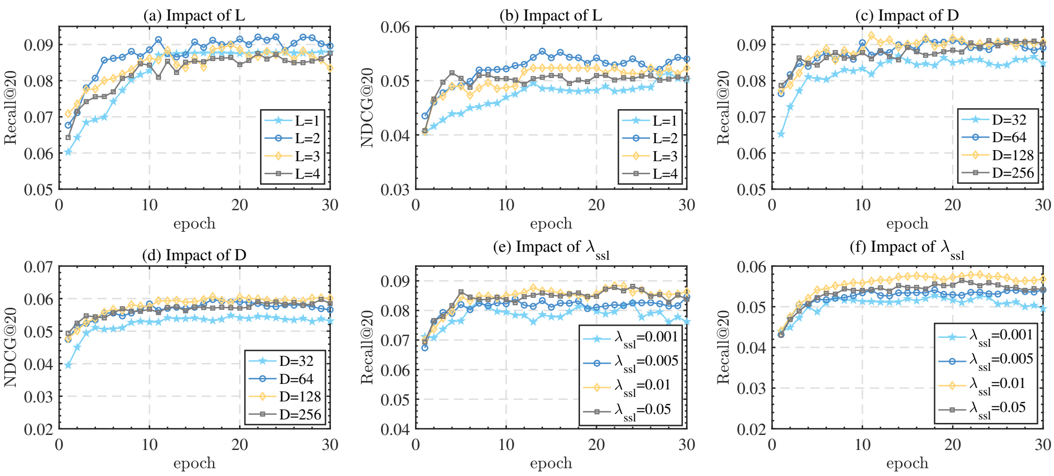

Figure 7. Impact of hyper-parameters on SHLKR Performance: L (**a**,**b**), D (**c**,**d**), and λ_{ssl} (**e**,**f**) on Yelp Dataset.

5.4. Ablation Study

To evaluate the effects of different components in SLHKR on model performance, we individually removed each key component and conducted an ablation study. By comparing the variant models to SLHKR, we can assess the contribution of different components to SLHKR's performance. The following provides a detailed description of these variant models:

- SLHKR-HG: This variant model removes the global hypergraph relation encoder module and relies only on the existing pairwise relationships to learn the local representations of users and items.
- SLHKR-KHG: This variant removes the hypergraph learning in the item channel and relies only on a single channel for learning user representation.
- SLHKR-SSL: This variant removes the self-supervised learning task and only considers the influence of the main task on the model optimization.

The results of the ablation experiment on three datasets are shown in Figure 8. From the results, it can be observed that each module contributes positively to the model's performance. When only using pairwise relationships, the performance significantly decreases compared to the complete model. These results confirmed the effectiveness of utilizing hypergraph structures to capture high-order relationships. This further validates that aggregating high-order semantic information from different hyperedge types can effectively improve the model's performance. Compared to the dual-channel learning, the contribution of the single channel is relatively small. But, it still brings improvements on model performance. This suggests that aggregating high-order information of all items based on hypergraph convolution can alleviate the impact of data sparsity on model performance, especially under sparse data conditions. Lastly, the self-supervised learning task provides additional benefits to the model's performance. One reason is that maximizing the mutual information between similar nodes and distinguishing different nodes in graph learning can reduce the influence of noise interactions on embedding learning. Simultaneously, maximizing the mutual information between different views can leverage supervised signals to complement representation learning.

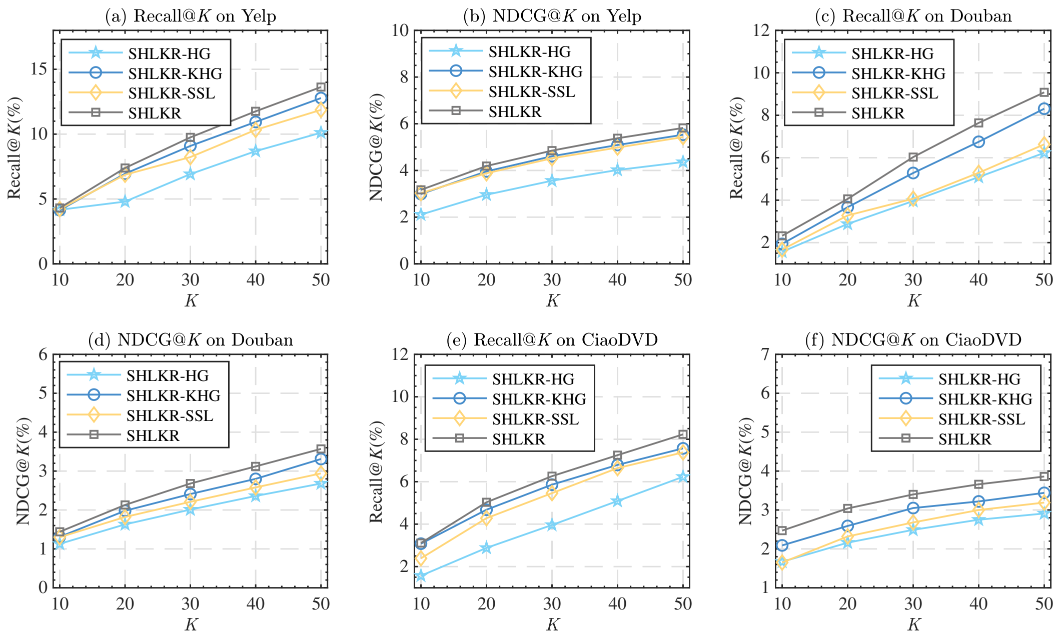

Figure 8. Performance comparisons of different variant models of SHLKR across three datasets: (**a**,**b**) Yelp, (**c**,**d**) Douban, and (**e**,**f**) CiaoDVD.

5.5. Effect of Multi-Type Hyperedges

To analyze the effects of different hyperedge types in the user and item channels on SLHKR's performance, we designed the following ablation experiments. We gradually removed different hyperedge types, generating three variants of SLHKR. The variant SHLKR-s has social context-based hyperedges in the user channel removed. This means that the model cannot leverage high-order relationships from social connections. SHLKR-b has the behavior-context-based hyperedges in both the user and item channels removed. This means that the model cannot utilize high-order relationships from user–item interaction behaviors. SHLKR-a has the attribute-context-based hyperedges eliminated. As a result, it loses the ability to leverage external knowledge derived from the attribute graph.

Figure 9 illustrates the effect of each hypergraph guided by different hyperedge types on the model's performance. From the results, we can draw the conclusion that each hypergraph structure contributes positively to the model's performance. This result demonstrates the effectiveness of hypergraphs in modeling high-order relationships. Typically, when the hyperedges generated from the social and behavior context are removed, the model's performance is most affected. However, the variants still outperform the scenario when only utilizing the observed graph. These high-order relationships can provide valuable data patterns for modeling users' preferences. Removing the hypergraph generated from the attribute context type has a relatively smaller impact on performance. While these attributes can provide valuable features, they may not capture the direct interactions between users and items.

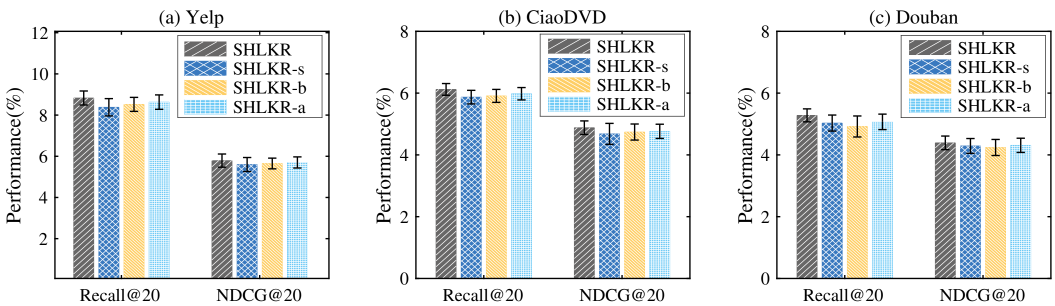

Figure 9. Effect of multi-type hyperedges on model performance with respect to Top@K across three datasets: (**a**) Yelp, (**b**) CiaoDVD, and (**c**) Douban.

6. Conclusions and Future Work

In this paper, we propose a self-supervised hypergraph learning framework for knowledge-aware social recommendation. It constructs hyperedges in hypergraphs by integrating three main types of connections: user behavior, social relationships, and attribute context. By employing a dual-channel hypergraph structure and hypergraph convolution, we effectively modeled high-order relationships between users and items, presenting a novel and powerful approach for knowledge-aware social recommendation systems. Additionally, we introduced a self-supervised learning task to mitigate the sensitivity to noisy feedback. Extensive experiments on public datasets have demonstrated the effectiveness of SHLKR. In our further research, we plan to utilize the timestamps of user–item interactions to encode temporal context and better understand the dynamic nature of user behavior. Specifically, we will consider the following temporal contexts: short-term dynamics, long-term preferences, and trends in popular items. By incorporating these temporal contexts, we aim to enhance the model's capability for modeling the dynamic changes in user interests. This enables SHLKR to better adapt to shifts in user preferences across temporal contexts.

Author Contributions: Conceptualization, M.L. and L.Y.; methodology, M.L.; software, M.L. and Q.D.; validation, M.L. and J.L.; resources, M.L.; data curation, M.L. and Q.D.; writing—original draft preparation, M.L.; writing—review and editing, M.L., J.L. and L.Y.; visualization, M.L.; supervision, L.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by JSPS KAKENHI (grant number 22KJ2935) and the Waseda University Grants for Special Research Projects (grant number 2023C-430).

Data Availability Statement: We conducted experiments on publicly available datasets. Below are the links for these datasets: Yelp/Douban: http://shichuan.org/dataset (accessed on 16 September 2023); CiaoDVD: https://github.com/SocialRecsys (accessed on 23 May 2023).

Conflicts of Interest: Qi Ding was employed by the company Rakuten. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Burke, R.; Felfernig, A.; Göker, M.H. Recommender systems: An overview. *Ai Mag.* **2011**, *32*, 13–18. [CrossRef]
2. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [CrossRef]
3. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* **2012**, arXiv:1205.2618.
4. Jamali, M.; Ester, M. A matrix factorization technique with trust propagation for recommendation in social networks. In Proceedings of the Fourth ACM Conference on Recommender Systems, Barcelona, Spain, 26–30 September 2010; pp. 135–142.
5. Ma, H.; King, I.; Lyu, M.R. Learning to recommend with social trust ensemble. In Proceedings of the 32nd international ACM SIGIR Conference on Research and Development in Information Retrieval, Boston, MA, USA, 19–23 July 2009; pp. 203–210.
6. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81. [CrossRef]
7. Wu, L.; Sun, P.; Hong, R.; Fu, Y.; Wang, X.; Wang, M. Socialgcn: An efficient graph convolutional network based model for social recommendation. *arXiv* **2018**, arXiv:1811.02815.
8. Wu, L.; Sun, P.; Fu, Y.; Hong, R.; Wang, X.; Wang, M. A neural influence diffusion model for social recommendation. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 235–244.
9. Gao, Y.; Zhang, Z.; Lin, H.; Zhao, X.; Du, S.; Zou, C. Hypergraph learning: Methods and practices. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 2548–2566. [CrossRef] [PubMed]
10. Gao, Y.; Feng, Y.; Ji, S.; Ji, R. HGNN+: General hypergraph neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 3181–3199. [CrossRef] [PubMed]
11. Zhu, Z.; Gao, C.; Chen, X.; Li, N.; Jin, D.; Li, Y. Inhomogeneous Social Recommendation with Hypergraph Convolutional Networks. *arXiv* **2021**, arXiv:2111.03344.
12. Yang, H.; Chen, H.; Li, L.; Philip, S.Y.; Xu, G. Hyper meta-path contrastive learning for multi-behavior recommendation. In Proceedings of the 2021 IEEE International Conference on Data Mining (ICDM), Auckland, New Zealand, 7–10 December 2021; pp. 787–796.

13. Ji, S.; Feng, Y.; Ji, R.; Zhao, X.; Tang, W.; Gao, Y. Dual channel hypergraph collaborative filtering. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 6–10 July 2020; pp. 2020–2029.
14. Yu, J.; Yin, H.; Li, J.; Wang, Q.; Hung, N.Q.V.; Zhang, X. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 413–424.
15. Long, X.; Huang, C.; Xu, Y.; Xu, H.; Dai, P.; Xia, L.; Bo, L. Social recommendation with self-supervised metagraph informax network. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Virtual Event, 1–5 November 2021; pp. 1160–1169.
16. Pan, X.; Gan, M. Multi-behavior recommendation based on intent learning. *Multimed. Syst.* **2023**, *29*, 3655–3668. [CrossRef]
17. Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; Xie, X. Self-supervised graph learning for recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 11–15 July 2021; pp. 726–735.
18. Zhang, C.; Chen, R.; Zhao, X.; Han, Q.; Li, L. Denoising and Prompt-Tuning for Multi-Behavior Recommendation. In Proceedings of the ACM Web Conference 2023, Austin, TX, USA, 30 April–4 May 2023; pp. 1355–1363.
19. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [CrossRef]
20. Wang, X.; He, X.; Wang, M.; Feng, F.; Chua, T.S. Neural graph collaborative filtering. In Proceedings of the 42nd international ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 165–174.
21. Wang, X.; Jin, H.; Zhang, A.; He, X.; Xu, T.; Chua, T.S. Disentangled graph collaborative filtering. In Proceedings of the 43rd international ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 25–30 July 2020; pp. 1001–1010.
22. Wang, X.; He, X.; Cao, Y.; Liu, M.; Chua, T.S. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 950–958.
23. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 25–30 July 2020; pp. 639–648.
24. Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, E.; Tang, J.; Yin, D. Graph neural networks for social recommendation. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 417–426.
25. Wu, Q.; Zhang, H.; Gao, X.; He, P.; Weng, P.; Gao, H.; Chen, G. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 2091–2102.
26. Shi, C.; Hu, B.; Zhao, W.X.; Philip, S.Y. Heterogeneous information network embedding for recommendation. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 357–370. [CrossRef]
27. Wang, J.; Ding, K.; Hong, L.; Liu, H.; Caverlee, J. Next-item recommendation with sequential hypergraphs. In Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, Virtual Event, 25–30 July 2020; pp. 1101–1110.
28. Han, J.; Tao, Q.; Tang, Y.; Xia, Y. DH-HGCN: dual homogeneity hypergraph convolutional network for multiple social recommendations. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 2190–2194.
29. Li, Y.; Gao, C.; Luo, H.; Jin, D.; Li, Y. Enhancing hypergraph neural networks with intent disentanglement for session-based recommendation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 1997–2002.
30. Xia, L.; Huang, C.; Zhang, C. Self-supervised hypergraph transformer for recommender systems. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 2100–2109.
31. Qiu, J.; Chen, Q.; Dong, Y.; Zhang, J.; Yang, H.; Ding, M.; Wang, K.; Tang, J. GCC: Graph contrastive coding for graph neural network pre-training. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 23–27 August 2020; pp. 1150–1160.
32. Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; Wang, L. Graph contrastive learning with adaptive augmentation. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 2069–2080.
33. Xu, D.; Cheng, W.; Luo, D.; Chen, H.; Zhang, X. InfoGCL: Information-aware graph contrastive learning. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 30414–30425.
34. Shen, X.; Sun, D.; Pan, S.; Zhou, X.; Yang, L.T. Neighbor contrastive learning on learnable graph augmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37; pp. 9782–9791.
35. You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; Shen, Y. Graph contrastive learning with augmentations. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 5812–5823.
36. Velickovic, P.; Fedus, W.; Hamilton, W.L.; Liò, P.; Bengio, Y.; Hjelm, R.D. Deep graph infomax. *ICLR (Poster)* **2019**, *2*, 4.
37. Yadati, N.; Nimishakavi, M.; Yadav, P.; Nitin, V.; Louis, A.; Talukdar, P. Hypergcn: A new method for training graph convolutional networks on hypergraphs. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1511–1522

38. Zheng, J.; Liu, J.; Shi, C.; Zhuang, F.; Li, J.; Wu, B. Dual similarity regularization for recommendation. In Proceedings of the Advances in Knowledge Discovery and Data Mining: 20th Pacific-Asia Conference, PAKDD 2016, Auckland, New Zealand, 19–22 April 2016; pp. 542–554.
39. Zhao, T.; McAuley, J.; King, I. Leveraging social connections to improve personalized ranking for collaborative filtering. In Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2014; pp. 261–270.
40. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.S. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth, WA, Australia, 3–7 April 2017; pp. 173–182.