

Article

Social Recommendation Algorithm Based on Self-Supervised Hypergraph Attention

Xiangdong Xu ¹, Krzysztof Przystupa ^{2,*}  and Orest Kochan ³ ¹ China Mobile Group Zhejiang Co., Ltd., Shaoxing Branch, Shaoxing 312000, China² Department of Automation, Lublin University of Technology, Nadbystrzycka 36, 20-618 Lublin, Poland³ School of Computer Science, Hubei University of Technology, Wuhan 430068, China

* Correspondence: k.przystupa@pollub.pl

Abstract: Social network information has been widely applied to traditional recommendations that have received significant attention in recent years. Most existing social recommendation models tend to use pairwise relationships to explore potential user preferences, but overlook the complexity of real-life interactions between users and the fact that user relationships may be higher order. These approaches also ignore the dynamic nature of friend influence, which leads the models to treat different friend influences equally in different ways. To address this, we propose a social recommendation algorithm that incorporates graph embedding and higher-order mutual information maximization based on the consideration of social consistency. Specifically, we use the graph attention model to build higher-order information among users for deeper mining of their behavioral patterns on the one hand; while on the other hand, it models user embedding based on the principle of social consistency to finally achieve finer-grained inference of user interests. In addition, to alleviate the problem of losing its own hierarchical information after fusing different levels of hypergraphs, we use self-supervised learning to construct auxiliary branches that fully enhance the rich information in the hypergraph. Experimental results conducted on two publicly available datasets show that the proposed model outperforms state-of-the-art methods.

Keywords: social recommendation; hypergraph; attention mechanism; mutual information



Citation: Xu, X.; Przystupa, K.; Kochan, O. Social Recommendation Algorithm Based on Self-Supervised Hypergraph Attention. *Electronics* **2023**, *12*, 906. <https://doi.org/10.3390/electronics12040906>

Academic Editors: Marco Lombardi, Domenico Santaniello, Zilu Liang and Muhammad Khan

Received: 3 December 2022

Revised: 1 February 2023

Accepted: 4 February 2023

Published: 10 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recommendation systems have had great success in various applications that predict how likely a user is to be interested in a particular item. Nevertheless, the data collection is often labor-intensive, which leads to a high risk of cold-start problems in recommendation [1,2]. Aiming to alleviate this problem, researchers have added various auxiliary information to recommender systems to improve their accuracy. This auxiliary information plays a significant role as side information of the user-item interaction, thus accurately mining the user's interests, for instance, tourism recommendation [3], food recommendation [4], etc. In this regard, fusing users' social information with user-project interaction data effectively improves recommendation performance. Consequently, in recent years, social recommendation algorithms use social information as auxiliary information, which is then used to outline a more complete user profile. This leads to personalized recommendations for users' interests, which has attracted the attention of an increasing number of researchers.

Generally, the social recommendation model uses the direct social relationships formed among users (such as friends, concerns and worries, user relevance, etc.) as auxiliary information to improve recommendation performance. On the basis of the matrix decomposition collaborative filtering method with the user-item interaction matrix as input, researchers tried to fuse the information of the user's social relationship matrix in different ways [5,6] to build a socialized recommendation model. Then users and items were mapped into

low-dimensional space, and embedded representations of users and items were obtained for subsequent prediction and recommendation. At present, many fruitful works have been developed by researchers in the field of social recommendation, thus effectively mining the influence of social friends on user preferences. Guo et al. [7] explore how to use the graph neural network (GNN) model to solve the problems of online data management and blurred preference feedback boundaries in social recommendation. Paolo et al. [8] proposed the use of trust metrics among users to improve the performance of recommendations, which can be used to replace the trust weights of similarity weights. Jamali et al. [9] presented a random wandering model that combines a trust relationship-based approach to model user social relationships with a collaborative filtering approach. Jamali et al. [10] introduced a social network recommendation approach based on matrix decomposition techniques. Chen et al. [11] suggested using an attention-based memory module fusing the aspect- and friend-level of users. Tao et al. [12] explored the social recommendation problem in depth and introduced knowledge distillation techniques in the model training.

The above works have achieved good research results, but the following problems are still a challenge:

1. Most research work in social recommendation assumes that all of the user's friends have the same influence on them or that they all have similar interests to them. This assumption is too far from the actual real-life situation due to the different types of friends. In real life, different friends have different influences on a user's decision-making [13]. As shown in Figure 1, user B has two friends; B and A both like to play volleyball and have the same preference for singing as C. B takes A's suggestion more into account when doing outdoor activities and C's suggestion more into account when doing indoor activities. Singing and playing volleyball, however, have different degrees of influence on the users. Chen et al. [14] proposed to fuse social influence between users with two attention networks for modeling complex dynamic and general static preferences of users in social recommendations. However, only the importance of friends is considered and the impact on users in different aspects is neglected.
2. GNN-based social recommendation models have shown outstanding performance [15–17]. Nevertheless, these models widely exploit only simple pairwise user relationships and overlook the complex higher-order relationships between users. Inspired by hypergraph learning [18], Yu et al. [19] used social information for constructing hypergraphs to model higher-order interactions through multi-channel hypergraph convolution. Problems in constructing multiple levels of hypergraph for fusion, however, cannot take into account the independence of different levels of hypergraph modeling itself.

To address the above problems, inspired by the social inconsistency modeling proposed by Yang et al. [20] In this paper, we develop a novel self-supervised social recommendation model called Hypid Graph Attention Hypergraph model (henceforth HGATH). Specifically, we use a hypergraph attention network to mine the implicit higher-order information in the original user embedding. The hypergraph is constructed by unifying the user–friend–items that form triangular relationships, and a multi-channel hypergraph convolutional attention model mines the hidden information between users. We aim to construct hypergraph models of different aspects of user attributes based on different channels representing different types of high-level relationships of users, which leads to the study's richer user representations with higher-order implicit information. Simultaneously, we propose a combination of sampling as well as relational attention mechanisms that highlight the different effects of different types of friends on the performance of the final recommendation results based on the idea of social inconsistency. In addition, inspired by self-supervised learning, in this paper, we exploit the ranked nature of the hypergraph structure to hierarchically maximize the mutual information among users, user-centric sub-hypergraphs, and global hypergraphs to achieve performance improvement. The experiment results show that HGATH has superior performance with respect

to mean absolute error (MAE), and root mean square error (RMSE) compared to current state-of-the-art methods.

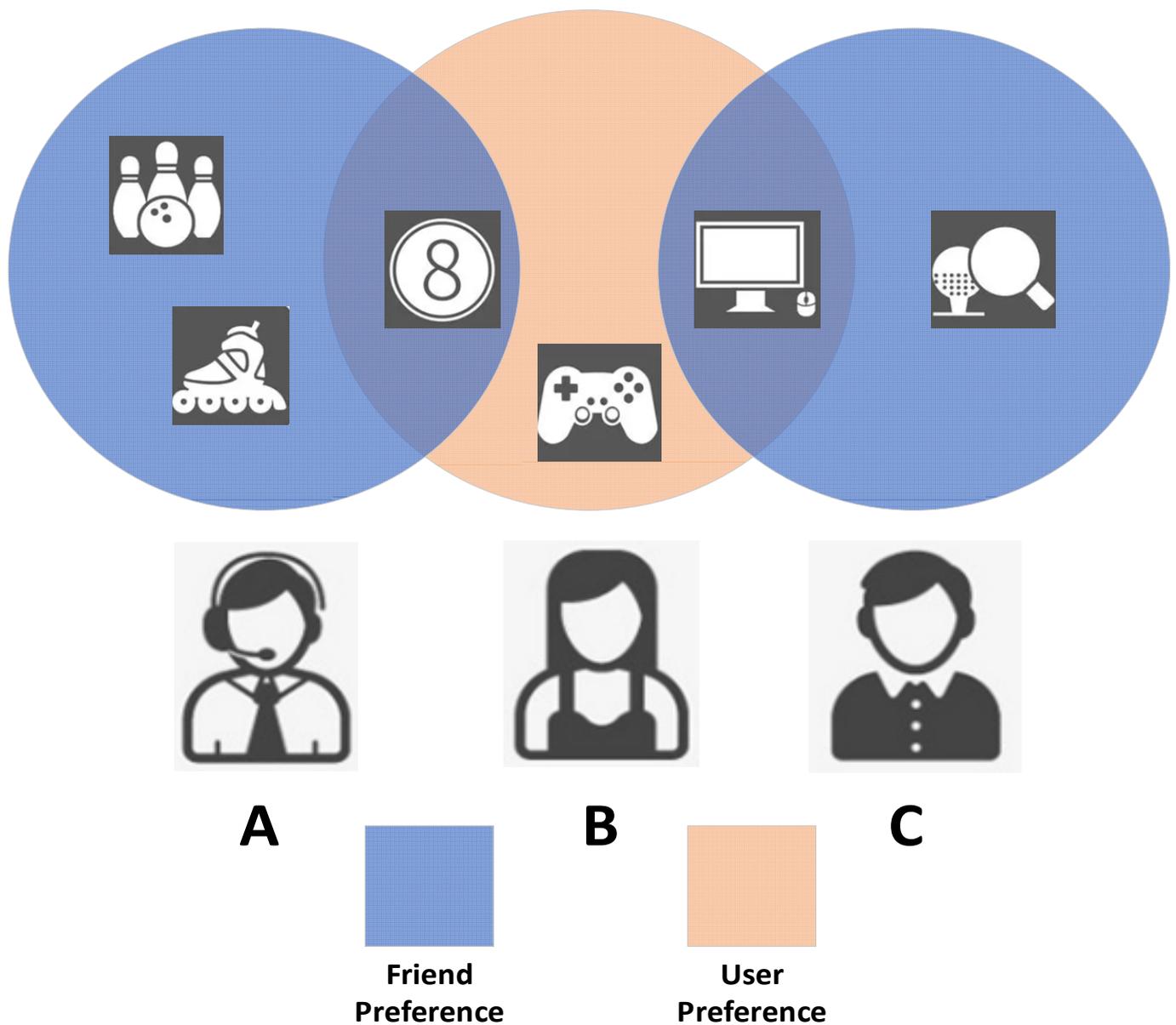


Figure 1. Typical scenario for social recommendation.

The contributions are as follows.

1. By modeling different levels of hypergraphs to enrich user feature expressions, and introducing hypergraphs to dig deeper into the higher-order information among users, we conclude with a description of the transcendent pairwise relationships among users through hypergraph modeling.
2. Based on the consideration of social consistency, a new attention framework is proposed to highlight the influence of different friends on the final recommendation results. This framework also deeply highlights the important influence of friends who make important suggestions in a certain aspect on the final recommendation results, which better simulates the real recommendation scenario.
3. We propose to integrate the self-supervised learning strategy seamlessly with hypergraph model training to enhance the model's performance.

4. Experiments on two publicly available datasets show that the proposed model HGATH outperforms state-of-the-art social recommendation models.

2. Related Works

In this section, we give a brief review of two lines of research related to our work: recommendation algorithms based on attention mechanisms and social recommendation algorithms based on GNNs.

2.1. Social Recommendation Based on Attention Mechanism

Social recommendation has been studied in recent years with the formation and development of social networks in various online services that use consumers' social information as auxiliary information to alleviate the problem of sparse interaction data in traditional recommendation systems to improve recommendation performance. Related sociological theories have also demonstrated that a user's preferences are similar to or easily influenced by those around him/her. Social recommendation based on matrix decomposition is one of the most commonly used methods because matrix decomposition methods have the advantages of accurate recommendation effect, good scalability, and low implementation complexity. The basic idea of these methods is to incorporate users' social relationship information grounded in collaborative filtering recommendation methods which are based on matrix decomposition, aimed at the mining of better potential feature vectors.

As the publication of the literature shows [21], researchers have started to consider the use of attention mechanisms to improve existing relevant social recommendation algorithms that have fruitful results [19–22]. Pei et al. [22] proposed a new interactive attention-gated recursive network for social recommendations to accurately capture their joint impact on user-item interactions and to better model the dynamics of user and item changes. In particular, the attention mechanism is innovatively applied to learn the attention scores of user and item histories, and thus explore the dynamic dependencies between users and items. Zhang et al. [23] proposed a new meta-path-based social recommendation model to strengthen the model performance by using meta-paths to simulate the user-item-society relationship, where a combination of attention mechanism and multigraph representation model was used to achieve the representation fusion of multigraph models. Yu et al. [24] proposed a multi-channel hypergraph network, which deeply explores the higher-order associations between users-items through hypergraph modeling, while employing an attention mechanism to efficiently aggregate embeddings modeled from multiple channels. Zhang et al. [25] proposed an inductive context personalization (ICP) framework based on context learning, in which a neural aggregator based on an attention network is introduced to fuse the heterogeneous content of entities for optimizing the sorting scheme by expressing the paired relationship between entities.

2.2. Social Recommendation Based on Graph Neural Networks

Deep learning has been successful in artificial intelligence and machine learning by bringing great progress to society; it is characterized by stacking multiple layers of neural networks, which then have better learning representation capabilities. Many scholars have applied deep learning techniques to the field of recommendation and have achieved good results [26,27].

Since GNNs establish a deep learning framework for graph structure, they can utilize both graph structure information and node feature information, and construct more complex and deeper neural networks for representation learning when compared with network representation learning methods, thus becoming a popular research topic in recent years. To simplify the problem and facilitate modeling, most of the early GNNs were based on simple graph structures, i.e., static, homogeneous pairwise graph structures. GraphRec [28] constructed the user, item, and social network as a graph model integrating node information and topology, then captured the interaction information between the graph models using a principled strategy based on the consideration of user social relationship hetero-

generality. DiffNet [29] and the follow-up work of the same team DiffNet++ [30] model the diffusion of users' social influence and potential interest based on the diffusion model and its theory. They constructed heterogeneous graphs of social and interest networks, digging deeper into higher-order social and interest networks, then aggregated the embeddings of the heterogeneous graphs based on a new attention mechanism to precisely learn the representations of nodes in the graph. Wu et al. [29] argued that it is the user's friends who are dynamic and multiple aspects influence the user's preferences. Therefore, they constructed two graph-embedding models to model the user and item domains, and then further aggregated the graph-embedding models based on the reinforcement learning and attention frameworks by adaptively adjusting the weighting influence of multifaceted graph embeddings. Song et al. [31] developed a new GNN social recommendation model that addresses the problems of ignoring potential personal interests hidden in user-item interactions and high computational costs in the social recommendation which is based on the influence of partially important social relationships, and uses diffusion theory to make the computation more efficient.

In the real world, the relationships between things are often not in pairs, but between two or more entities that together form an interactive relationship [32]. Using simple graphs to represent such non-pairwise relationships would result in information loss [33]. Hypergraphs extend the definition of simple graphs. In the hypergraph, a hyperedge can contain any number of nodes, so it can store non-pairwise relationships directly. Hypergraphs have more flexible edge definitions than simple graphs, and thus are more powerful for representing complex relationships.

The hypergraph neural network (HGNN) [34] was the first work to design hypergraph convolution operations from a spectral perspective to deal with complex data correlations in representation learning. Tan et al. [35] pioneered the study of user alignment in social networks using a hypergraph structure, which differs from the traditional approach in which a hypergraph approach is used to model higher-order relationships. Ji et al. [36] developed a jump hypergraph convolutional collaborative filtering solution based on a two-channel learning strategy, which exploits the hypergraph structure to deeply explore the higher-order relationships between users and items, while using a two-channel strategy to flexibly characterize users and items. In addition, hybrid matrix factorization (HMF) [37] combines matrix decomposition techniques and hypergraph networks in the field of social recommendation; this exploits the hypergraph structure to model contextual relationships while comprehensively exploring the role of contextual information in the recommendation process based on the matrix decomposition model. Location based social networks (LBSN2Vec) [35] focus on the relationship between user mobility and social relationships, and thus propose specialized hypergraph embedding methods for inferring user preferences, which combine hypergraphs and random wandering strategies for learning node representations of hypergraphs based on sampling in terms of user check-ins and social relationships. Comparison between HGATH and the competing methods are given in Table 1.

Table 1. Comparison between HGATH and other related works.

Approaches	Details	Gaps
Attention network-based modeling	The combination of attention mechanisms with different neural network models has yielded fruitful results in social recommendation models, including gated neural network approaches [22], meta-paths [23], and especially joint work with hypergraphs [24] and the application of multilayer attention networks in social recommendation models [25].	User representation modeling

Table 1. Cont.

Approaches	Details	Gaps
Simple graph-based modeling	Different graph neural network models have achieved fruitful results in social recommendation models, including the graph convolution approach [28], graph diffusion approach [29–31], etc.	User representation modeling based on different diffusion ideas
Hypergraph-based modeling	The application of hypergraphs in social recommendation models has yielded fruitful results, for example, in the literature [34–36], especially work on the combination of hypergraphs and matrix decomposition techniques [37] and the combination of hypergraphs and random roaming strategies in social recommendation models [38].	High-order relations between users and complex relations between users and items

Our work is different from the previous approaches. First, we construct different levels of hypergraphs based on the actual situation of realistic scenarios, thus obtaining a richer representation of user features. Second, based on the starting point of social consistency, we propose a framework that integrates sampling and attention mechanisms to accurately model the importance of users' different friends.

3. Self-Supervised Hypergraph Attention Recommendation Model

In this section, the proposed self-supervised hypergraph attention model HGATH is introduced in detail, and the overall structure of the model is illustrated in Figure 2 below. HGATH model has three submodules. The first is the social consistency module, which aims to extract friends who have higher consistency with users. The second submodule is the hypergraph attention module, which is designed to model higher-level connections between users. The third is a self-supervised learning module, which alleviates the problem of data sparsity in social recommendation. Specifically, first in the social consistency module, we transform the input user data $U = \{u_1, u_2, \dots, u_m\}$ and item data $\mathcal{T} = \{t_1, t_2, \dots, t_n\}$ into user and item embeddings, respectively, afterwards. Then, we unite the two obtained embeddings together to generate a query embedding to select consistent neighbors. Next, we perform the selection using a neighbor sampling strategy, in which the sampling probability is based on the consistency score between the query embedding and the neighbor embedding. After sampling, the neighbors with consistent relationships are given higher weights for aggregation using relational attention, and finally the consistency-processed user embeddings U_c and item embeddings T_c are obtained. In this way, the influence of different friends on the user is highlighted, while the importance of friends who give essential advice in a certain area to the user is also strongly highlighted. Meanwhile, in the hypergraph attention module, according to $U = \{u_1, u_2, \dots, u_m\}$, we use the semantic hypergraphs constructed for different aspects and the graph attention network to learn the higher-order relations of user information, thus presenting key user semantic information. Finally, we use hypergraph encoded user information combined with user embeddings modeled utilizing a consistency policy to obtain more comprehensive user representations U_T . Then, the obtained accurate user embedding is fed into the self-supervised learning module. For self-supervised tasks, further enhanced user representation is achieved by hierarchically maximizing the information based on the interaction between different aspects of the user representation. Comprehensive user representations, which were considered to contain multiple types of higher-order relational information, were obtained based on a self-supervised task. Finally, the recommendation list was obtained by an inner product operation with the precise item embeddings obtained in the social consistency module.

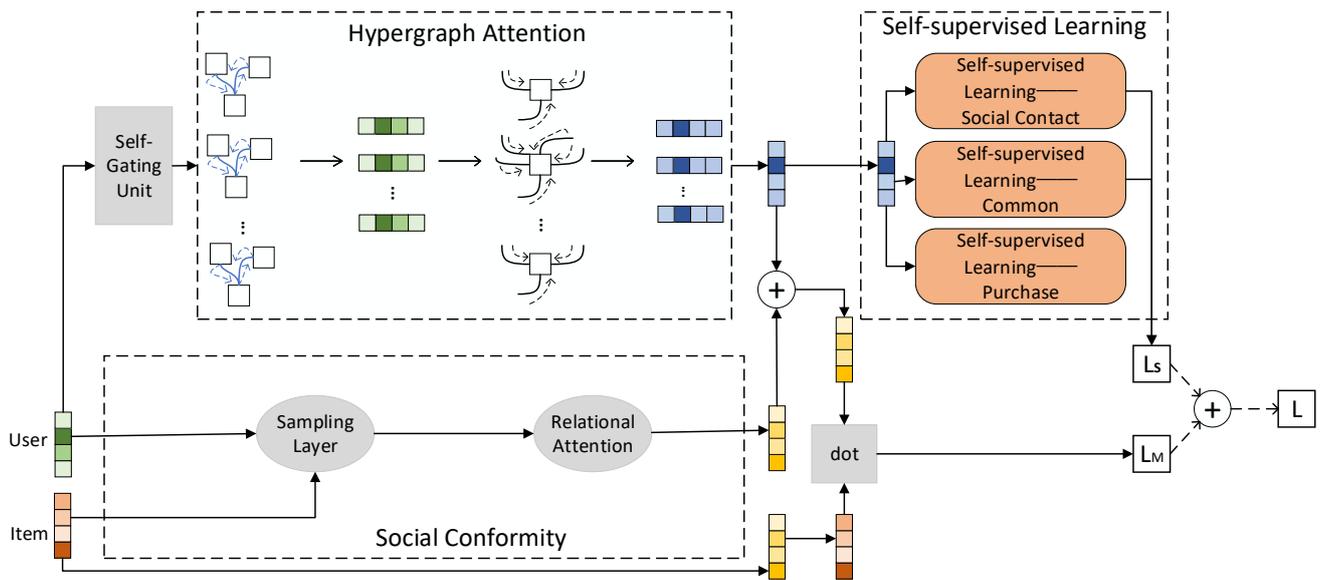


Figure 2. The framework of HGATH.

To facilitate the understanding of the forthcoming text and formulae we introduce the nomenclature of symbols used in this paper in Table 2.

Table 2. The nomenclature of mathematical symbols used in this paper.

Symbol	Description
$U = \{u_1, u_2, \dots, u_m\}$	User data.
$\mathcal{T} = \{t_1, t_2, \dots, t_n\}$	Item data.
U_T, e_u	User embedding.
T_c, e_t	Item embedding.
$q_{u,t}$	Query embedding.
$e_u, e_t \in \mathbb{R}^d$	Embedding of the nodes u and t .
ψ	ReLU activation function.
$\mathbf{W}^{(l)} \in \mathbb{R}^{2d \times d}$	Encoding function.
$h_v^{(0)}$	Initial embedding of node v .
$o^{(l)}(i; b)$	Node embedding of node i at layer l .
$\alpha_i^{(l)}$	Importance of the i -th neighbor sampled.
Q	Sum of the neighbors sampled.
r_i	Relationship of edge (v, i) .
$e_{r_i} \in \mathbb{R}^d$	Relational embedding of relation r_i .
$\mathbf{w}_a \in \mathbb{R}^{2d}$	Trainable parameter.
a_i	Attention weight.
$h_u^{(L)}$	Embeddings of u .
$h_t^{(L)}$	Embeddings of t .
$E_u^{(0)}$	Original user embedding.
$E_t^{(0)}$	Original item embedding.
$W_g^c \in \mathbb{R}^{d \times d}$	Training weight parameter.

Table 2. Cont.

Symbol	Description
$b_g^c \in \mathbb{R}^d$	Bias parameter.
c	Different hypergraph channels.
σ	Sigmoid function.
W_f	Trainable weight matrix.
x_k^{l-1}	Feature information of node k in the l th layer of the neural network.
a_1^T	Weight parameter.
u_k	Correlation of node K on the hyperedge e_j .
a_{jk}	Different weights when super-edge aggregation.
h_i^l	Updated feature information of node v_i .
W_h	Weight matrix.
b_{ij}	Attention coefficient of node v_i on the hyperedge e_j .
a_2^T	Weight parameter.
v_j	Correlation of super-edges at node i .
H_t	User embedding vector after averaging.
$E_t^{(l)}$	Potential feature vector of the user.
A_c	Adjacency matrix.
a_u^c	Row vector of A_c .
$sum(a_u^c)$	Number of the sub-hypergraph.
$\eta_{out_2} : \mathbb{R}^{m \times d} \rightarrow \mathbb{R}^d$	Readout function.
$f_D(\cdot) : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$	Discriminant function.
\mathcal{E}_{rating}	Rating error (RMSE) between \mathcal{E}_{rating} and the true value (u, t) for all $\hat{R}_{u,t}$ pairs in $R_{u,t}$.
\hat{y}_i	Predicted value.
y_i	True value.

3.1. Social Consistency

Wu et al. [29] utilize embedding technology for the first step of item and user formalization. Inspired by the work [29], $U = \{u_1, u_2, \dots, u_m\}$ and $\mathcal{T} = \{t_1, t_2, \dots, t_n\}$ are first transformed into user embeddings e_u and item embeddings e_t ,

$$q_{u,t} = \sigma\left(W_q^T (e_u \oplus e_t)\right) \quad (1)$$

where $q_{u,t}$ is the query embedding and $e_u, e_t \in \mathbb{R}^d$ is the embedding of the nodes u and t . This part is designed with a query layer that dynamically samples the neighbors according to different items. The embedding obtained after node aggregation is as follows:

$$\mathbf{h}_v^{(l)} = \psi\left(\mathbf{W}^{(l)T} \left(\mathbf{h}_v^{(l-1)} \oplus \text{AGG}^{(l)} \left\{ \mathbf{h}_i^{(l-1)} \mid i \in \mathcal{N}_v \right\}\right)\right) \quad (2)$$

where ψ is the ReLU activation function, $\mathbf{W}^{(l)} \in \mathbb{R}^{2d \times d}$ is the encoding function, and $h_v^{(0)}$ is the initial embedding of node v . Moreover, it is recommended to emphasize more consistent

rather than inconsistent neighbors when aggregating them. Therefore, the consistency score across all neighbors is as follows:

$$p^{(l)}(i; b) = o^{(l)}(i; b) / \sum_{j \in \mathcal{N}_v} o^{(l)}(j; b) \tag{3}$$

where $o^{(l)}(i; b)$ denotes the node embedding of node I at layer l .

Thus, we utilize relational attention to learn the importance of these sampled nodes. Relational attention attributes a significant factor to each sampled node i . Then the AGG function in (2) can be reformulated as follows:

$$\text{AGG}^{(l)} = \sum_{i=1}^Q \alpha_i^{(l)} \cdot h_i^{(l-1)} \tag{4}$$

where $\alpha_i^{(l)}$ denotes the importance of the i -th neighbor sampled and Q denotes the sum of the neighbors sampled. Assuming that the relationship of edge (v, i) is r_i , the attention formula for α_i is obtained as:

$$\alpha_i^{(l)} = \frac{\exp(\mathbf{w}_a^\top (\mathbf{h}_i^{(l-1)} \oplus \mathbf{e}_{r_i}))}{\sum_{j=1}^Q \exp(\mathbf{w}_a^\top (\mathbf{h}_j^{(l-1)} \oplus \mathbf{e}_{r_j}))} \tag{5}$$

where $\mathbf{e}_{r_i} \in \mathbb{R}^d$ denotes the relational embedding of relation r_i , $\mathbf{w}_a \in \mathbb{R}^{2d}$ is the trainable parameter, and a_i is the attention weight.

Following L-layer propagation, the embeddings of u and t can be obtained, denoted as $h_u^{(L)}$ and $h_t^{(L)}$, respectively.

3.2. Hypergraph Attention Network

The hypergraph was first used to construct user-item and social networks. In this paper, similar to the literature [19], we also designed a multi-channel hypergraph network to represent users in many aspects. Inspired by the literature [39], we also used a gate network to process the user embedding and separate it into different channels; the original user embedding $E_u^{(0)}$ is defined as follows. Meanwhile, we generated the original item embedding $E_t^{(0)}$ based on the same principle.

$$E_u^{(0)} = f_{gate}^c(e_u) = e_u \odot \sigma(e_u W_g^c + b_g^c) \tag{6}$$

$$E_t^{(0)} = f_{gate}^c(e_t) = e_t \odot \sigma(e_t W_g^c + b_g^c) \tag{7}$$

In (6), $W_g^c \in \mathbb{R}^{d \times d}$ and $b_g^c \in \mathbb{R}^d$ denote the training weight parameter and bias parameter, respectively, c refers to different hypergraph channels, and e_u and e_t are the initial user embedding and item embedding, respectively.

We used two different attention mechanisms through two aggregation processes to get the final representation.

We first learned the hyperedge representation of node based on the attention mechanism in combination with hypergraph attention (HyperGAT) [40] for characterized node feature.

$$f_j^l = \sigma \left(\sum_{v_k \in e_j} a_{jk} W_f x_k^{l-1} \right) \tag{8}$$

$$X^{(0)} = g(E_u^{(0)} \oplus E_t^{(0)}) \tag{9}$$

$$a_{jk} = \frac{\exp(a_1^T u_k)}{\sum_{v_p \in e_j} \exp(a_1^T u_p)} \tag{10}$$

$$u_k = \text{LeakyReLU}(W_1 x_k^{l-1}) \tag{11}$$

where σ is the Sigmoid function, W_f is a trainable weight matrix, and x_k^{l-1} denotes the feature information of node k in the l th layer of the neural network. Equation (9) show the fusion of user information and item information. a_1^T is the weight parameter and u_k denotes the correlation of node K on the hyperedge e_j . a_{jk} highlights the effect of different weights when there is super-edge aggregation.

Next, all hyperedges are denoted as $\{f_j^l \mid \forall e_j \in E_i\}$. We used the high-order relationship of users based on hyperedge modeling to learn the representation of nodes

$$h_i^l = \sigma \left(\sum_{e_j \in E_i} b_{ij} W_h f_j^l \right) \tag{12}$$

where h_i^l is the updated feature information of node v_i , W_h represents the weight matrix, and b_{ij} indicates the attention coefficient of node v_i on the hyperedge e_j , as follows:

$$b_{ij} = \frac{\exp(a_2^T v_j)}{\sum_{e_p \in E_i} \exp(a_2^T v_p)} \tag{13}$$

$$v_j = \text{LeakyReLU} \left(\left[W_2 f_j^l \parallel W_f h_i^{l-1} \right] \right) \tag{14}$$

where, a_2^T is the weight parameter, v_j denotes the correlation of super-edges at node i .

The final-user representation of the hypergraph is as follows:

$$H_t = \frac{1}{L+1} \sum_{l=0}^L E_t^{(l)} \tag{15}$$

where H_t denotes the user embedding vector after averaging and $E_t^{(l)}$ refers to the potential feature vector of the user.

3.3. Self-Supervised Learning

Inspired by the literature [16], self-supervised learning was added to the training of the model in order to improve the performance. The three semantic hypergraphs capture higher-order information using the adjacency matrix A_c . Each row in A_c represents a subgraph of the corresponding hypergraph with the user represented by the index of that row at its center. Thus a hierarchical structure can be derived based on the relationship between hypergraphs and sub-hypergraphs: user nodes—user-centered sub-hypergraphs—hypergraphs, and from this structure self-supervised signals are created. We designed a new function $\eta_{out_1} : \mathbb{R}^{m \times d} \rightarrow \mathbb{R}^d$, as follows:

$$z_u^c = \eta_{out_1}(P_c, a_u^c) = \frac{P_c a_u^c}{\text{sum}(a_u^c)} \tag{16}$$

where $P_c = f_{gate}^c(P)$ is to control P . We set a_u^c to represent the row vector of A_c the sub-graph. $\text{sum}(a_u^c)$ denotes the number of the sub-hypergraph. Similarly, another readout function $\eta_{out_2} : \mathbb{R}^{m \times d} \rightarrow \mathbb{R}^d$ is defined, which is actually an average set that summarizes the obtained sub-hypergraph embedding:

$$h^c = \eta_{out_2}(Z_c) = \text{AveragePooling}(Z_c) \tag{17}$$

In accordance with deep graph infomax (DGI), we used info noise contrastive estimation (InfoNCE) as the learning objective using hierarchical mutual information maximization. As a result, the objective function of the self-supervised task is defined as follows.

$$\mathcal{L}_s = - \sum_{c \in \{s,j,p\}} \left\{ \sum_{u \in U} \log \sigma(f_D(p_u^c, z_u^c) - f_D(p_u^c, \tilde{z}_u^c)) + \sum_{u \in U} \log \sigma(f_D(z_u^c, h^c) - f_D(\tilde{z}_u^c, h^c)) \right\} \tag{18}$$

where $f_D(\cdot) : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ is a discriminant function. We break Z_c by shuffling between rows and columns to create negative examples of \tilde{Z}_c .

3.4. Model Optimization

For the user representation $\mathbf{h}_i^{(L)}$, the score is obtained by summing and averaging the H_t and calculating the inner product with the item representation $\mathbf{h}_t^{(L)}$

$$\hat{R}_{u,t} = Avg(\mathbf{h}_u^{(L)} + H_t) \cdot \mathbf{h}_t^{(L)} \tag{19}$$

where $Avg(\cdot)$ is the mean function and $\hat{R}_{u,t}$ is the scoring score. The primary loss function for the recommendation task is referred to as the root mean square error (RMSE) between \mathcal{E}_{rating} and the true value (u, t) for all $\hat{R}_{u,t}$ pairs in $R_{u,t}$:

$$\mathcal{L}_m = \sqrt{\frac{\sum_{(u,t) \in \mathcal{E}_{rating}} (R_{u,t} - \hat{R}_{u,t})^2}{|\mathcal{E}_{rating}|}} \tag{20}$$

Thus, the total loss consists of two parts: one is the loss of recommended tasks and the other is the loss of self-supervised learning tasks, as follows:

$$\mathcal{L} = \mathcal{L}_m + \beta \mathcal{L}_s \tag{21}$$

where β adjusts two loss functions.

4. Experiment Comparison and Analysis

4.1. Dataset

We conducted experiments on two public datasets, Epinions [13] and Ciao [41]. Epinions is an online social network that provides an item evaluation and review service to allow other users to add to their own trust list, indicating the approval of this user’s ratings and reviews. The dataset contains a list of ratings given to items by users, as well as the social relationships between users and trusted users. The Ciao dataset is similar to Epinions with the exception of the point in time when the trust relationship was established, but it captures the text content of each comment and information about the comment from other users. To preprocess the data, we delete users with non-social connections because they are not present in the social referrals, while linking neighboring items that share more than 50%. The datasets’ statistical parameters are summarized in Table 3.

Table 3. Statistical information of the datasets.

Dataset	Ciao	Epinions
#of Users	6776	15,210
#of Items	101,415	233,929
#of Interactions	271,573	644,715
Interaction Density	0.0395%	0.0181%

4.2. Evaluation Criteria

To evaluate the performance of the algorithms used, we selected two evaluation metrics frequently used in recommendations—mean absolute error (MAE) and root mean square error (RMSE) [42], both of which are often used for performance analysis [43]. The formulas for their calculation are:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (22)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (23)$$

where \hat{y}_i is the predicted value and y_i is the true value.

4.3. Experimental Setting

We verified the efficiency of this model from the following aspects.

1. We considered a comparison with mainstream advanced algorithms to verify the efficiency and advancement of the algorithm proposed in this module.
2. We compared this model with graph neural networks to verify the effectiveness of the employed hypergraph attention.
3. We considered potential dimensionality factor analysis to test the effective dimensionality in various cases based on differing potential dimensionality factors to verify the robustness of the algorithm proposed in this module.

First, seven social recommendation algorithms proposed in recent years were selected for comparison.

1. SoRec [44] combined social information with matrix decomposition.
2. SocialMF [10] proposed a new social recommendation model, which is based on the matrix decomposition model and integrates the user's trust propagation mechanism to model social networks.
3. SoReg [44] proposed a recommendation algorithm based on social relations, which constrained the learning of users' potential feature vectors in the traditional matrix decomposition by social relations in a way of social regularization, so as to make the potential feature vectors of two users with social relations as similar as possible.
4. The collaborative user network embedding (CUNE) [45] proposed to dig deep into the implicit social relationship of users based on user feedback to identify the implicit important friends of users, and eventually realized the ranking of user preferences based on the Matrix Factorization (MF) ranking model.
5. Graph Convolution Matrix Completion with Spectral Nonlinear (GCMC + SN) [46] proposed a bipartite interactive graph-based graph autoencoder framework for differentiable message passing based on graph structure data.
6. Graph Recommendation (GraphRec) [28] developed a new graph neural network framework for social recommendation, which is the first time that a graph neural network was combined with a principled method to model users and projects and the complex interaction between them.
7. ConsisRec [15] used a sampling strategy to mine complex relationships among neighbors based on the principle of social consistency, thus modeling neighbors, and uses an attention mechanism to highlight the influence weights of different important users.

Second, we compared graph convolution, graph attention, and hypergraph convolution with hypergraph attention to verify the effectiveness of the hypergraph attention used in this paper. Finally, this model tested the impact of this algorithm under different latent dimensions and verified the robustness of the algorithm exhibited by the latent dimensions employed in this paper.

The dataset was randomly divided into training, validation, and test sets according to 60%, 20%, and 20% probability. The validation set was used to adjust the hyperparameters, and the predicted data from the training set was compared with the test set to obtain the final results. Meanwhile, we searched for percentages between 0.2, 0.4, 0.6, 0.8, and 1.0. For the embedding size, we adjusted between 8, 16, 32, 64, 128, and 256. The learning rate was adjusted between 0.0005, 0.001, 0.005, 0.01, 0.05, and 0.1 for batch sizes ranging from 32, 64, 128, and 256.

4.4. Experimental Analysis

4.4.1. Performance Comparison

The experimental results of all compared algorithms on the two datasets are shown in Figure 3 above. We observed the following conclusions:

- (a) On both datasets, CUNE, ConsisRec, and HGATH algorithms work better than SoRec, SocialMF, and SoReg, which shows the superior function of considering the consistency of social relationships shown on the model.
- (b) Among the social recommendation-based algorithms, HGATH outperforms the CUNE and ConsisRec algorithms, which indicates that for social recommendation, it is not enough to extract valid social relationships from users' social relationships. Obviously, the higher-order relationships among users must be considered. Meanwhile, compared with the simple graph structure that can only connect two nodes, the inset attention focuses on more complex relationships and mines richer user interaction information.
- (c) HGATH is more effective than GCMC + SN and GraphRec, which means that higher-order information interactions can maximize the inheritance of different aspects of hypergraph information.
- (d) On all the evaluation metrics, all comparison algorithms perform better on the Ciao dataset than the Epinions dataset; this is because the Epinions dataset is sparser than the Ciao dataset.

We adopted Recall@K and Normalized Discounted Cumulative Gain@K (NDCG@K) to evaluate the performance of all methods, where $K = 5, 10$. These two metrics have been widely used in previous recommendation studies to validate the accuracy. For each dataset, we used 80% as a training set to learn the parameters, 10% as a validation set to tune the hyperparameters, and 10% as a test set. We also followed similar experimental settings in the literature [39].

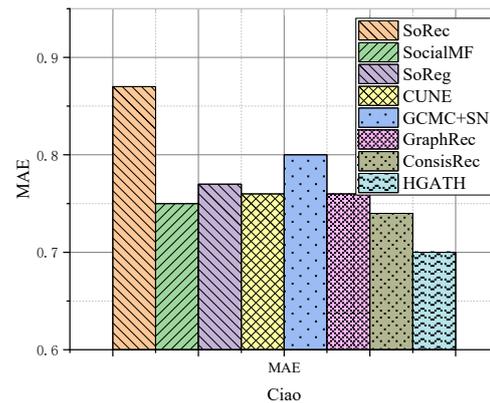
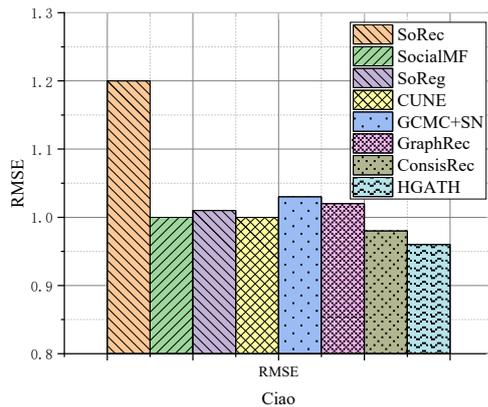
The performance of different methods on the Epinion and Ciao datasets are considered in Tables 4 and 5, respectively.

Table 4. Comparisons of different methods on the Epinion dataset.

	SoRec	SocialMF	SoReg	CUNE	GCMC + SN	GraphRec	ConsisRec	HGATH
Recal@5	0.217	0.215	0.221	0.233	0.241	0.249	0.255	0.261
Recal@10	0.259	0.263	0.257	0.271	0.282	0.287	0.297	0.323
NDCG@5	0.183	0.187	0.178	0.192	0.211	0.209	0.212	0.224
NDCG@10	0.198	0.207	0.211	0.228	0.241	0.238	0.249	0.267

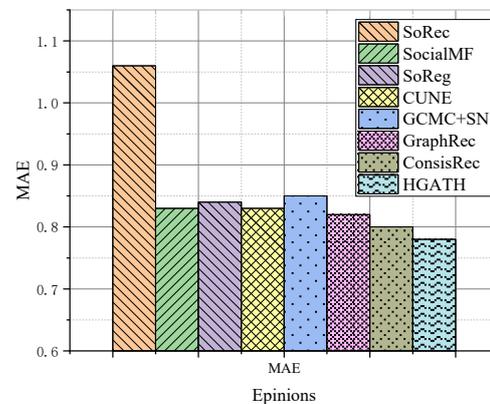
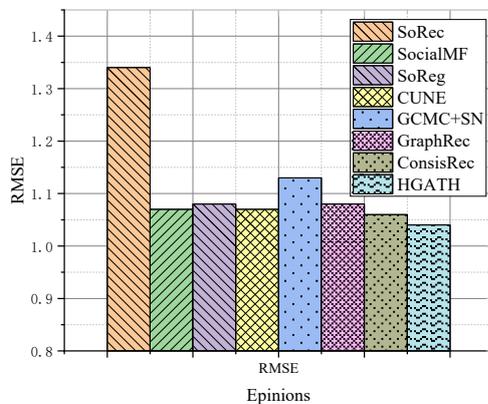
Table 5. Comparisons of different methods on the Ciao dataset.

	SoRec	SocialMF	SoReg	CUNE	GCMC + SN	GraphRec	ConsisRec	HGATH
Recall@5	0.229	0.235	0.234	0.244	0.257	0.261	0.271	0.271
Recall@10	0.274	0.277	0.275	0.291	0.322	0.318	0.326	0.347
NDCG@5	0.186	0.191	0.188	0.201	0.222	0.226	0.237	0.243
NDCG@10	0.208	0.211	0.215	0.233	0.251	0.256	0.266	0.275



(a) RMSE based on Ciao dataset HGATH compared with other state-of-the-art algorithms

(b) MAE based on Ciao dataset HGATH compared with other state-of-the-art algorithms



(c) RMSE based on a comparison of the Epinions dataset HGATH compared with other state-of-the-art algorithms

(d) MAE based on a comparison of the Epinions dataset HGATH compared with other state-of-the-art algorithms

Figure 3. The experimental results of all compared algorithms on the two datasets. (a) RMSE based on Ciao dataset HGATH compared with other state-of-the-art algorithms (b) MAE based on Ciao dataset HGATH compared with other state-of-the-art algorithms (c) RMSE based on a comparison of the Epinions dataset HGATH compared with other state-of-the-art algorithms (d) MAE based on a comparison of the Epinions dataset HGATH compared with other state-of-the-art algorithms.

In this section, we verify whether HGATH outperforms the baselines. We obtained the following observations.

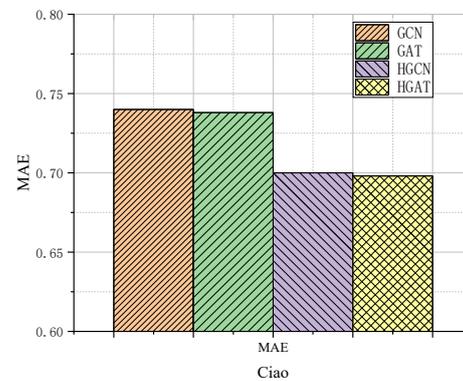
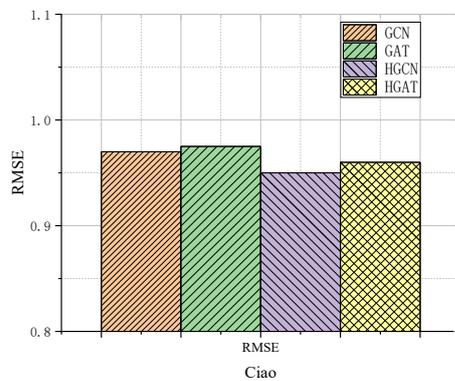
1. HGATH beats all baselines, which shows the effectiveness of the proposed method in this paper.
2. The graph model-based recommendation algorithms (e.g., GCMC + SN, GraphRec) outperform the social recommendation algorithms that do not employ graph learning

methods (e.g., SoRec, SocialMF, SoReg, CUNE), which validates the powerful learning capability of graph models in recommendation scenarios.

3. The performance difference between SoRec, SocialMF, and SoReg is not significant, which also validates that all three models are essentially MF-based recommendation models. The same reasoning applies to GCMC + SN and GraphRec.
4. The performance of each model based on the Epinion dataset is generally slightly lower than that of the Ciao dataset, which is consistent with the fact that the Ciao dataset is not as sparse as the Epinion one.

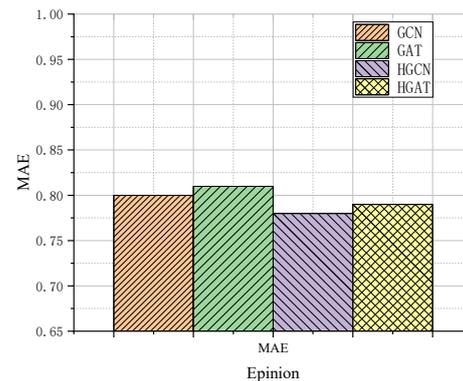
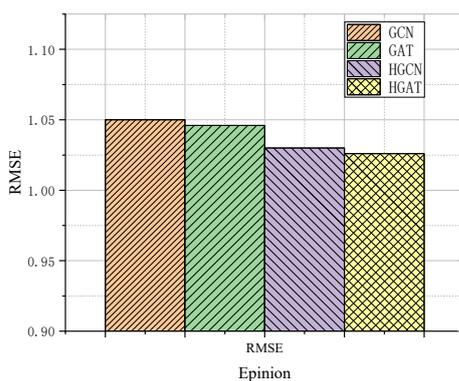
4.4.2. Comparative Analysis of Graph Neural Networks

To validate the presence or absence of graph neural network structure and the different effects brought by different graph neural network structures on the algorithm, this paper presents experiments and comparative analysis of graph convolution, graph attention, hypergraph convolution, and hypergraph attention. The experimental results are shown in the following Figure 4. The graph convolution network (GCN), graph attention (GAT), hypergraph convolution (HGCN), and hypergraph attention (HGAT) are presented in Figure 4.



(a) RMSE for the comparison of different graph neural networks based on the Ciao dataset

(b) MAE for the comparison of different graph neural networks based on the Ciao dataset



(c) RMSE for the comparison of different graph neural networks based on Epinions dataset

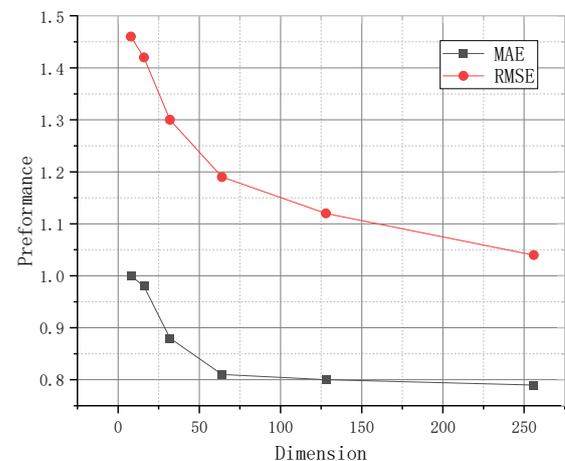
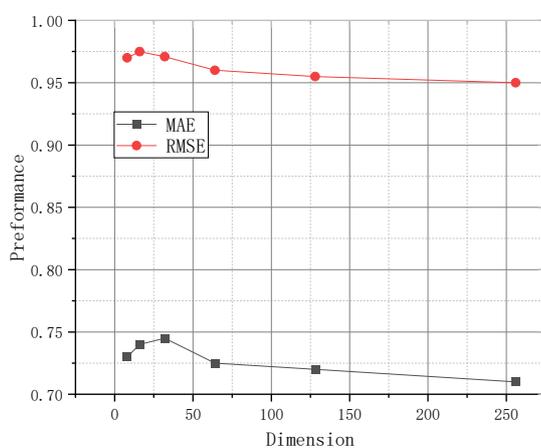
(d) MAE for comparison of different graph neural networks based on Epinions dataset

Figure 4. Experiments and comparative analysis of graph convolution, graph attention, hypergraph convolution, and hypergraph attention (a) RMSE for the comparison of different graph neural networks based on the Ciao dataset (b) MAE for the comparison of different graph neural networks based on the Ciao dataset (c) RMSE for the comparison of different graph neural networks based on the Epinions dataset (d) MAE for the comparison of different graph neural networks based on the Epinions dataset.

First, HGATH shows better performance in the two datasets. Moreover, the structure created using the hypergraph is obviously better than the simple graph structure, which proves the effectiveness of the hypergraph structure constructed in this paper for higher-order information extraction. Second, the difference between hypergraph convolution and hypergraph attention for two different metrics is not very obvious in the different datasets, which shows that although they are different hypergraph structures, they have their strengths and weaknesses in dealing with data with sparsity.

4.4.3. Potential Dimensional Analysis

The variation of potential dimensions also has a certain degree of influence on the experimental results, which are compared and analyzed to observe the adaptability of different potential dimensions to the algorithm. The experimental results are shown in Figure 5 below.



(a) Potential dimensional analysis of HGATH algorithm based on Ciao dataset

(b) Potential dimensionality analysis of HGATH algorithm based on Epinions dataset

Figure 5. The adaptability of different potential dimensions to the algorithm. (a) Potential dimensional analysis of HGATH algorithm based on Ciao dataset (b) Potential dimensionality analysis of HGATH algorithm based on Epinions dataset.

As shown by the experimental results, the effect becomes more pronounced when the dimensionality is larger, whereas the relative increase in the potential dimensionality increases memory usage and computing time. It is therefore obvious that the improvement of the algorithm levels off when the potential dimension increases to a certain level. Thus, the potential dimension 256 used in this paper is the optimal choice after comprehensive consideration.

5. Discussion

In this section, we summarize the results of the experiments and presents the discussion of the contributions made by our work as below:

1. In contrast to other social recommendation work that uses simple graph models to learn user representations, we used hypergraphs to learn user representations. The hypergraphs were used to learn the representation of the user based on multiple aspects of the user. As a complex graph, the hypergraph has the property of linking multiple nodes within one edge, which makes it naturally advantageous for the representation of complex data relationships between nodes. By taking advantage of the supergraph's superior learning ability to fully learn the user's representations, the complex relationships between users are well described, while higher-order relation-

ships are accurately modeled. The results in Figures 3 and 4 verify the effectiveness and efficiency of the method in this paper.

2. Users are influenced differently by different friends when making decisions. Instead of using the vanilla attention mechanism to distinguish the importance of friends, we designed a hierarchical iterative aggregation to learn the relational attention network of user representations. First, the user's friend features are dynamically sampled in a single layer based on item features, and then a dynamic aggregation operation is performed based on a relational attention mechanism to learn the user's representation. Next, each layer takes the previous layer's user's embedding as input, which finally outputs the user's iterative updated embedding. The whole process simulates how the potential embeddings of the users evolve with the dynamic influence of the project until the precise user embeddings are finally generated. In the process of simulated evolution, the attention network proposed in this paper accurately manifests the social consistency principle.
3. In order to further contextualize the effect of data sparsity while fully inheriting the rich user representation learned from the hypergraph, we innovatively incorporated self-supervised learning into the training of the recommendation model proposed in this paper. By considering the hypergraphs reflecting different aspects of user representations as different views in the self-supervised contrastive learning considered, the mutual information of these views was maximized to achieve rich user representations in the recommendation task for better performance.
4. Self-supervised learning is a fresh direction in the future recommendation field. However, graph learning-based recommendation models in self-supervised learning tasks mostly arbitrarily employ operations such as item cropping and masking to improve the variability among views in the self-supervised comparative learning process. Such operations also bring about the problem of creating more sparse training data. In the future, we will further investigate how to perform robust self-supervised learning while preserving the original data.

6. Conclusions

Considering social information as auxiliary information provides a big help to the recommendation system. Hence, we view social consistency as the first basis for considering social relationships in real scenarios and used hypergraph attention to model higher-order relationships among users when extracting special higher-order user information features. Simultaneously, higher-order mutual information maximization was used to alleviate the problem of higher-order information loss caused by fusing different aspects of hypergraph information. The experiments on two real datasets verify that the proposed algorithm outperformed the state-of-the-social recommendation algorithms. Further, in the future, we will consider how to simulate the changing interaction information in realistic scenarios based on dynamic graph models about the impact of friends on users.

Author Contributions: All authors contributed to the study conception and design. Conceptualization, X.X., K.P. and O.K.; methodology, X.X.; software, K.P.; validation, O.K.; formal analysis, O.K.; investigation, X.X.; data curation, K.P.; writing—original draft preparation, X.X. and O.K.; writing—review and editing, K.P.; visualization; K.P.; supervision, O.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financed as part of the Lublin University of Technology project: FD-20/EE-2/801 and project: FD-20/IM-5/087.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Forouzandeh, S.; Aghdam, A.; Forouzandeh, S.; Xu, S. Addressing the cold-start problem using data mining techniques and improving recommender systems by cuckoo algorithm: A case study of Facebook. *Comput. Sci. Eng.* **2020**, *40*, 62–73. [\[CrossRef\]](#)
2. Beshley, M.; Veselý, P.; Pryslupskyi, A.; Beshley, H.; Kyryk, M.; Romanchuk, V.; Kahalo, I. Customer-Oriented Quality of Service Management Method for the Future Intent-Based Networking. *Appl. Sci.* **2020**, *10*, 8223. [\[CrossRef\]](#)
3. Forouzandeh, S.; Rostami, M.; Berahmand, K. A hybrid method for recommendation systems based on tourism with an evolutionary algorithm and topsis model. *Fuzzy Inf. Eng.* **2022**, *24*, 26–40. [\[CrossRef\]](#)
4. Rostami, M.; Muhammad, U.; Forouzandeh, S.; Berahmand, K.; Farrahi, V.; Oussalah, M. An effective explainable food recommendation using deep image clustering and community detection. *Intell. Syst. Appl.* **2022**, *16*, 200157. [\[CrossRef\]](#)
5. De Meo, P.; Fotia, L.; Messina, F.; Rosaci, D.; Sarné, G.M. Providing recommendations in social networks by integrating local and global reputation. *Inf. Syst.* **2018**, *78*, 58–67. [\[CrossRef\]](#)
6. Przystupa, K.; Beshley, M.; Hordiichuk-Bublivska, O.; Kyryk, M.; Beshley, H.; Pyrih, J.; Selech, J. Distributed Singular Value Decomposition Method for Fast Data Processing in Recommendation Systems. *Energies* **2021**, *14*, 2284. [\[CrossRef\]](#)
7. Guo, Z.; Yu, K.; Li, Y.; Srivastava, G.; Lin, J.C.-W. Deep learning-embedded social internet of things for ambiguity-aware social recommendations. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 1067–1081. [\[CrossRef\]](#)
8. Massa, P.; Avesani, P. Trust-aware recommender systems. In Proceedings of the 2007 ACM Conference on Recommender Systems, Minneapolis, MN, USA, 19–20 October 2007; pp. 17–24.
9. Jamali, M.; Ester, M. Trustwalker: A random walk model for combining trust-based and item-based recommendation. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28–1 July 2009; pp. 397–406.
10. 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.
11. Chen, C.; Zhang, M.; Liu, Y.; Ma, S. Social attentional memory network: Modeling aspect- and friend-level differences in recommendation. In Proceedings of the 20th ACM International Conference on Web Search and Data Mining, Melbourne, Australia, 11–15 February 2019; pp. 177–185.
12. Tao, Y.; Li, Y.; Zhang, S.; Hou, Z.; Wu, Z. Revisiting graph based social recommendation: A Distillation Enhanced Social Graph Network. In Proceedings of the World Wide ACM Web Conference, Taipei, Taiwan, 20–24 April 2020; pp. 2830–2838.
13. Abu-Salih, B.; Wongthongtham, P.; Zhu, D.; Chan, K.Y.; Rudra, A. *Social Big Data Analytics*; Springer: Singapore, 2021.
14. Sun, P.; Wu, L.; Wang, M. Attentive recurrent social recommendation. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 185–194.
15. Xu, F.; Lian, J.; Han, Z.; Li, Y.; Xu, Y.; Xie, X. Relation-aware graph convolutional networks for agent-initiated social E-commerce recommendation. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 529–538.
16. Wu, J.; Fan, W.; Chen, J.; Liu, S.; Li, Q.; Tang, K. Disentangled contrastive learning for social recommendation. In Proceedings of the 31st ACM International Conference on Information and Knowledge Management, Atlanta, GA, USA, 17–21 October 2022; pp. 4570–4574.
17. Du, J.; Ye, Z.; Yao, L.; Guo, B.; Yu, Z. Socially-aware dual contrastive learning for cold-start recommendation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 1927–1932.
18. Yang, Y.; Huang, C.; Xia, L.; Liang, Y.; Yu, Y.; Li, C. Multi-behavior hypergraph-enhanced transformer for sequential recommendation. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 2263–2274.
19. 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 World Wide Web Conference 2021, Ljubljana, Slovenia, 12–16 April 2021; pp. 413–424.
20. Yang, L.; Liu, Z.; Dou, Y.; Ma, J.; Yu, P.S. Consisrec: Enhancing gnn for social recommendation via consistent neighbor aggregation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, Canada, 11–15 July 2021; pp. 2141–2145.
21. Ashish, V.; Noam, S.; Niki, P. Attention is All you Need. In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 1–11.
22. Pei, W.; Yang, J.; Sun, Z.; Zhang, J.; Bozzon, A.; Tax, D.M. Interacting attention-gated recurrent networks for recommendation. In Proceedings of the 2017 ACM Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1459–1468.
23. Zhang, C.; Wang, Y.; Zhu, L.; Song, J.; Yin, H. Multi-graph heterogeneous interaction fusion for social recommendation. *ACM Trans. Inf. Syst.* **2022**, *4*, 1–26. [\[CrossRef\]](#)
24. Yu, J.; Yin, H.; Li, J.; Gao, M.; Huang, Z.; Cui, L. Enhancing social recommendation with adversarial graph convolutional networks. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 3727–3739. [\[CrossRef\]](#)
25. Zhang, C.; Yao, H.; Yu, L. Inductive contextual relation learning for personalization. *ACM Trans. Inf. Syst.* **2022**, *39*, 1–22. [\[CrossRef\]](#)

26. Xia, L.; Huang, C.; Xu, Y.; Dai, P.; Zhang, X.; Yang, H.; Pei, J.; Bo, L. Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In Proceedings of the 35th AAAI Conference on Artificial Intelligence, Virtually, 22 February–1 March 2022; pp. 4486–4493.
27. Xia, L.; Xu, Y.; Huang, C.; Dai, P.; Bo, L. Graph meta network for multi-behavior recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtually, Canada, 11–15 July 2021; pp. 757–766.
28. 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 May 2019; pp. 417–426.
29. 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.
30. Wu, L.; Li, J.; Sun, P.; Hong, R.; Ge, Y.; Wang, M. Diffnet++: A neural influence and interest diffusion network for social recommendation. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 4753–4766. [[CrossRef](#)]
31. Song, W.; Xiao, Z.; Wang, Y.; Charlin, L.; Zhang, M.; Tang, J. Session-based social recommendation via dynamic graph attention networks. In Proceedings of the 20th ACM International Conference on Web Search and Data Mining, Melbourne, Australia, 11–15 February 2019; pp. 555–563.
32. 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.
33. Jin, B.; Cheng, K.; Zhang, L.; Fu, Y.; Yin, M.; Jiang, L. Partial relationship aware influence diffusion via a multi-channel encoding scheme for social Recommendation. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Galway, Ireland, 19–23 October 2020; pp. 585–594.
34. Feng, Y.; You, H.; Zhang, Z.; Ji, R.; Gao, Y. Hypergraph neural networks. In Proceedings of the 33th AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 3558–3565.
35. Tan, S.; Guan, Z.; Cai, D.; Qin, X.; Bu, J.; Chen, C. Mapping users across networks by manifold alignment on hypergraph. In Proceedings of the 28th AAAI Conference on Artificial Intelligence, Quebec, QC, Canada, 27–31 July 2014; pp. 159–165.
36. Ji, S.; Feng, Y.; Ji, R.; Zhao, X.; Tang, W.; Gao, Y. Dual channel hypergraph collaborative filtering. In Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, 6–10 July 2020; pp. 2020–2029.
37. Zheng, X.; Luo, Y.; Sun, L.; Ding, X.; Zhang, J. A novel social network hybrid recommender system based on hypergraph topologic structure. *World Wide Web J.* **2018**, *21*, 985–1013. [[CrossRef](#)]
38. Yang, D.; Qu, B.; Yang, J.; Cudre-Mauroux, P. Revisiting user mobility and social relationships in lbsns: A hypergraph embedding approach. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13 May 2019; pp. 2147–2157.
39. Chai, Y.; Jin, S.; Hou, X. Highway transformer: Self-gating enhanced self-attentive networks. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6887–6900.
40. Ding, K.; Wang, J.; Li, J.; Li, D.; Liu, H. Be more with less: Hypergraph attention networks for inductive text classification. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Virtual, 16–20 November 2020; pp. 4927–4936.
41. Ma, H.; Yang, H.; Lyu, M.R.; King, I. SoRec: Social recommendation using probabilistic matrix factorization. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, CA, USA, 26–30 October 2008; pp. 931–940.
42. Fang, M.-T.; Przystupa, K.; Chen, Z.-J.; Li, T.; Majka, M.; Kochan, O. Examination of Abnormal Behavior Detection Based on Improved YOLOv3. *Electronics* **2021**, *10*, 197. [[CrossRef](#)]
43. Jun, S.; Przystupa, K.; Beshley, M.; Kochan, O.; Beshley, H.; Klymash, M.; Wang, J.; Pieniak, D. A Cost-Efficient Software Based Router and Traffic Generator for Simulation and Testing of IP Network. *Electronics* **2020**, *9*, 40. [[CrossRef](#)]
44. Ma, H.; Zhou, D.; Liu, C.; Lyu, M.R.; King, I. Recommender systems with social regularization. In Proceedings of the 4th ACM International Conference on Web Search and Data Mining, Hong Kong, 9–12 February 2011; pp. 287–296.
45. Zhang, C.; Yu, L.; Wang, Y.; Shah, C.; Zhang, X. Collaborative user network embedding for social recommender systems. In Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, TX, USA, 27–29 April 2017; pp. 381–389.
46. Liu, X.; He, J.; Duddy, S.; O’Sullivan, L. Convolution-consistent collective matrix completion. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2209–2212.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.