User-Context Collaboration and Tensor Factorization for GNN-Based Social Recommendation

Wei Wang, Zhenzhen Quan, Siwen Zhao, Guoqiang Sun, Yujun Li, *Member, IEEE*, Xianye Ben, *Senior Member, IEEE*, and Jianli Zhao

***Abstract*—One goal of social recommendation is to utilize social information to alleviate data sparsity and improve recommendation accuracy. User social relationships are inherently graph-structured, graph neural network (GNN) has recently attracted extensive attention in social recommendation because of its capability to integrate structural information and topology. However, current graph neural network (GNN)-based social recommendation models fail to consider context information during user interactions, which hinders more accurate modeling of user interest features. To address this problem, we propose a new social recommendation model based on context-aware graph neural network named CENTRIC (User-ContExt CollaboratioN and TensoR Factorization for GNN-based SocIal ReCommendation). Specifically, first a multi-channel GNN model with user-context collaboration module is designed, so that context can directly affect user features and participate in the calculation of user interaction probability with items. Then tensor factorization is adopted in output layer to effectively fuse the features extracted from different channels. Experiments on three public datasets show that CENTRIC significantly outperforms other state-of-the-art social recommendation models, further experiments also demonstrate that context information and tensor factorization help improve the accuracy of GNN-based social recommendation.**

***Index Terms*—Social recommendation, context-aware, graph neural network, tensor factorization.**

I. Introduction

The core of recommender system is to accurately model user preferences based on user-item historical interactions, thereby predicting users' interest in items [1], [2], [3], [4]. One of the challenges of recommender system is the serious data sparsity problem in real datasets. Social recommendation [5], [6], [7] seeks to use social networks as additional information to assist in modeling user features. According to social relationship theory [8], [9], users with close connections on social networks tend to have more similar preferences and can influence each other's decisions. Ma et al. proposed the SoRec model [10], which was the earliest study on matrix-based social recommendation. After that, they proposed a social regularization recommendation model SoReg [11], which assumed that user feature vector should be similar to friends' feature vectors. Considering that the influence of implicit feedback was ignored in the previous works, Guo et al. introduced social trust information to reconstruct the user feature matrix on the basis of SVD++, and proposed TrustSVD [12]. Recently, Zhao et al. extended matrix-based social recommendation to higher-order tensors and proposed a tensor factorization-based social recommendation model named TrustTF [13].

User social networks and user-item interaction data essentially have graph structure. In recent years, some studies have introduced the popular graph neural network (GNN) technology to model social recommendation algorithms. Benefiting from the superiority of processing structured data, GNN [14], [15], [16] can naturally and explicitly encode key cooperative signals (i.e. topology), propagate and aggregate information from neighbor nodes to enhance user and item feature representations [17], [18]. Compared with traditional machine learning methods, GNN-based recommendation models often achieve higher recommendation accuracy, because they can mine high-order interactive information of network nodes and further alleviate data sparsity. Rianne et al. applied GNN to model user-item bipartite graph and proposed GC-MC [19], which applied graph convolution to recommendation for the first time. Then, Wang et al. extended GC-MC to multiple layers and proposed the NGCF model [20]. Wu et al. constructed a GCN-based social influence diffusion model DiffNet [21]. Then they considered user-item collaboration information based on DiffNet and proposed the enhanced DiffNet++ [22]. Fu et al. designed a dual side deep context-aware modulation module named DICER, using both social information and user-item interaction history to enhance the feature representation of users and items [23].

On the other hand, extensive researches have shown that user decisions are also affected by additional factors such as time, location and mood, which are called context. For example, the types of shows people watch at different times of the day may vary. Our previous work and some related researches [13],

Manuscript received 3 November 2022; revised 14 February 2023; accepted 14 March 2023. Date of publication 27 March 2023; date of current version 25 October 2023. This work was supported in part by the National Key R&D Program of China under Grant 2020YFC0833200, in part by the Natural Science Foundation of China under Grants 61971468 and 62271289, in part by the Natural Science Fund for Outstanding Young Scholars of Shandong Province under Grant ZR2022YQ60, in part by the Major Project of Independent Innovation in Qingdao under Grant 21-1-2-18-xx and in part by the Natural Science Foundation of Shandong Province under Grant ZR2021MF104. Recommended for acceptance by Dr. Zhen Wang. *(Corresponding authors: Yujun Li; Xianye Ben; Jianli Zhao.)*

Wei Wang, Zhenzhen Quan, Siwen Zhao, Guoqiang Sun, Yujun Li, and Xianye Ben are with the School of Information Science and Engineering, Shandong University, Qingdao 266237, China (e-mail: 202120421@mail.sdu.edu.cn; quanzhenzhen@mail.sdu.edu.cn; 202012766@mail.sdu.edu.cn; 202020377@mail.sdu.edu.cn; liyujun@sdu.edu.cn; benxianye@126.com).

Jianli Zhao is with the School of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China (e-mail: jlzhao@sdust.edu.cn).

Digital Object Identifier 10.1109/TNSE.2023.3258427

[24], [25] have demonstrated that context information can help improve the accuracy and generality of personalized recommender systems. Alexandros et al. first proposed a context-aware collaborative filtering model based on tensor factorization [26], which was an early approach to context modeling. Baltrunas et al. proposed an item segmentation method based on context pre-filtering and combined it with the traditional matrix factorization algorithm [27]. Considering that some scenarios may have a large variety of contexts and it is difficult to select the most helpful context for recommendation, Wu et al. proposed a bias tensor factorization model BiasTF, which used regression tree-based autoencoders to choose part of contexts for users [28]. Recently, some works such as xDeepFM [29] and GCM [30] also extended FM to neural networks to model context features.

However, to the best of our knowledge, current GNN-based social recommendation models fail to consider context information and explicitly simulate the influence of context on user features, which hinders the further improvement of recommendation accuracy. Although recent DICER [23] mentioned the concept of context-aware, it only took the user preferences and item attributes enhanced by the graph as the context without additional information input. Due to the lack of context information, the user features in current GNN-based models are only affected by long-term factors such as user-item interaction records and social networks, while the short-term factors such as time and location are ignored. This prevents more accurate modeling of user preference.

To solve this problem, this paper proposes a new GNN-based social recommendation model with user-context collaboration and tensor factorization named CENTRIC (User-**C**ont**E**xt Collaboratio**N** and **T**enso**R** Factorization for GNN-based Soc**I**al Re**C**ommendation). The model considers three graph structures simultaneously: social graph, user-context bipartite graph, user-item interaction graph, and includes four channels: social channel, user-context collaborative channel, users collaborative channel and items collaborative channel. Specifically, we first construct a multi-channel GNN model with user-context collaborative filtering module, which directly aggregates the features of neighbor contexts in user features, and the context features are also affected by neighbor users. As user features and context features are updated alternately, the model will capture more user-context collaboration signals, thus enhancing the representation of users. Furthermore, the introduction of context information makes the matrix factorization (MF) of traditional output layer no longer applicable, so we propose a tensor factorization fusion method instead of matrix factorization or multilayer perceptron (MLP) commonly used in other researches. As a powerful tool for processing higher-order data, tensor factorization can effectively capture user-item-context interactions and predict the probability of users interacting with items under different contextual conditions. Finally, we conduct multiple experiments on three well-known public datasets. The experimental results verify the effectiveness of considering context in GNN-based social recommendation, and the recommendation accuracy of CENTRIC is higher than other state-of-the-art social recommendation models.

To sum up, the main contributions of this paper are as follows:

1) We propose a new GNN-based social recommendation model with user-context collaboration and tensor factorization, which can effectively capture additional user-context collaboration signals to better model user features.
2) We propose a feature fusion method based on tensor factorization at the output layer to effectively fuse the features extracted from different channels.

The rest of this paper is organized as follows: Section II introduces the three graph structures in CENTRIC and the main notations used. Section III introduces the network structure and feature fusion method of CENTRIC in detail. In Section IV, we present three research questions and introduce the experimental setup. In Section V, we analyze the experimental results and verify the superiority of CENTRIC. Finally, we summarize our work and introduce the future research plan.

II. Graph Construction and Notations

By loading the user-item-context interaction records and social trust data, the model can obtain the user-user social graph $G^S = (V_U, E^S_{UU})$, user-item interaction graph $G^R = (V_U \cup V_I, E^R_{UI})$ and user-context interaction graph $G^C = (V_U \cup V_C, E^C_{UC})$. The social graph G^S shows the social connections E^S_{UU} among user nodes V_U, and the features of trusted friends can be directly used to enhance the features of target users. G^R shows the historical interaction records E^R_{UI} between user nodes V_U and item nodes V_I, we can further calculate the similar user graph $S_U = (V_U, E^{Sim}_{UU})$ and the similar item graph $S_I = (V_I, E^{\mathrm{Sim}}_{II})$, where E^{Sim}_{UU} denotes connected similar users, and E^{Sim}_{II} denotes connected similar items. G^C shows the interaction records E^C_{UC} between user nodes V_U and context nodes V_C, the context can also be used to assist in the calculation of user similarity and enhance user features.

Assuming that the graphs contain M users, N items and K contexts, the model initializes the user embedding matrix $P^{M\times D}$, the item embedding matrix $Q^{N\times D}$ and the context embedding matrix $C^{K\times D}$, where D denotes the embedding dimension. For each user node u, item node i and context node c, this study defines initial user feature vector p^0_u, initial item feature vector q^0_i, initial context feature vector c^0_k, and output them to the multi-channel aggregation part. The main mathematical notations used in this paper are summarized in Table I.

III. The Proposed Model

This section first introduces the overview of CENTRIC (see III-A). Then the multi-channel aggregation method is introduced in detail (see III-B). Feature fusion based on tensor factorization and model training methods are also presented (see III-C and III-D). Finally, the space complexity and time complexity of CENTRIC are discussed (see III-E).

A. Overview

As shown in Fig. 1, CENTRIC includes four parts: graph construction, multi-channel aggregation, incorporate implicit feedback and output layer. First, the model inputs three original

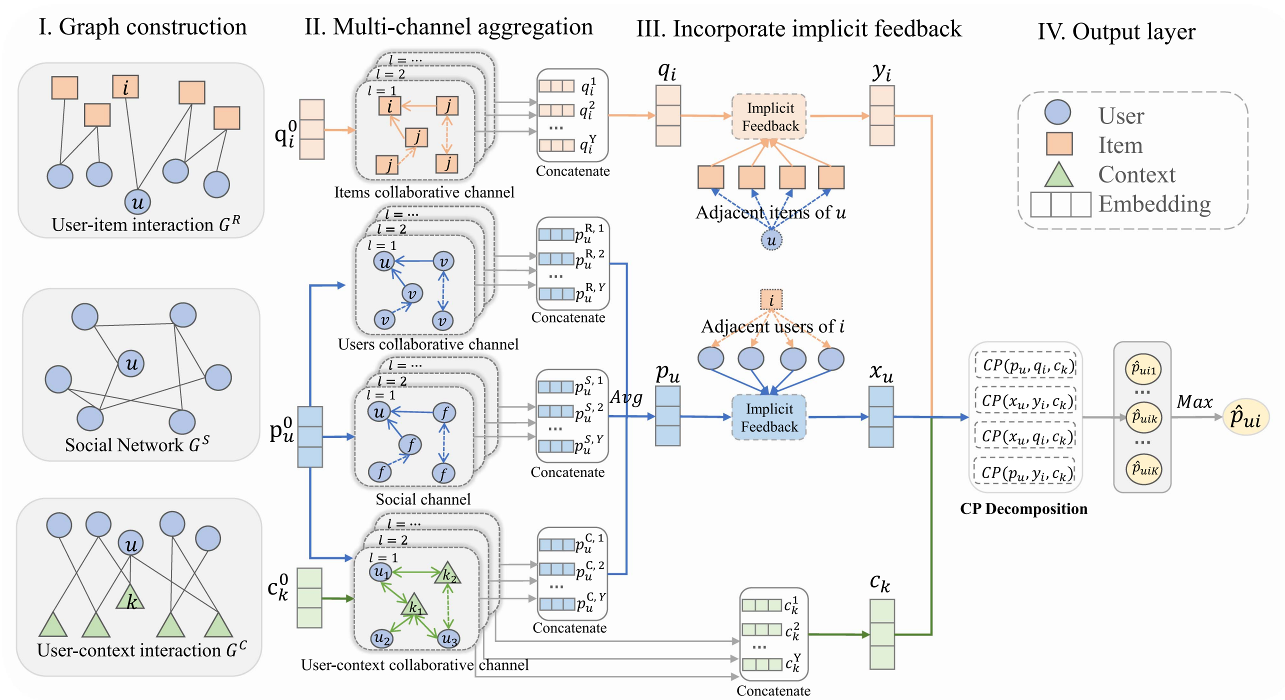


Fig. 1. Overview of the CENTRIC model. I) The model inputs social graph, user-item interaction graph, and user-context interaction graph. II) The information transfer in the three graphs is constructed as four channels, and the model outputs the graph enhanced user feature p_u, item feature q_i, and context feature c_k. III) To further alleviate data sparsity, implicit feedback is considered in user feature and item feature. IV) Finally, the output layer applies tensor factorization to fuse user-item-context features.

TABLE I
SUMMARY OF MAIN NOTATIONS

Notations	Definitions
G^S	user-user social graph
G^R	user-item interaction graph
G^C	user-context interaction graph
S_U	similar user graph
S_I	similar item graph
$p_u^{S,l}$	the l-th layer feature vector of user u in graph G^S
$p_u^{C,l}$	the l-th layer feature vector of user u in graph G^C
$p_u^{R,l}$	the l-th layer feature vector of user u in graph S_U
q_i^l	the l-th layer feature vector of item i in graph S_I
c_k^l	the l-th layer feature vector of context k in graph G^C
x_u^i	feature vector of user u with implicit feedback
y_i^u	feature vector of item i with implicit feedback
T_u	set of trusted friends of user u
$G^U(u)$	set of contexts interacting with user u
$G^C(k)$	set of users interacting with context k
$S_U(u)$	set of similar friends of user u
$S_I(i)$	set of similar items of item i
$R_U(i)$	set of users who interacted with item i
$R_I(u)$	set of items that have been interacted with by user u
t_{uf}	social weight of user u with friend f
s_{uv}	similarity coefficient of user u with user v
s_{ij}	similarity coefficient of item i with item j
D	embedding dimension
W	the weight matrix in neural network
Y	depth of neural network

graph structures: user-item interaction graph, user-user social graph, and user-context interaction graph. Then user features, item features and context features aggregate the high-order neighborhood information in the graph respectively through four channels. In order to further enhance the feature representation, the model also considers the influence of implicit feedback. Finally, tensor factorization is used at the output layer to calculate the probability of user interacting with item. The following sections describe how CENTRIC works in detail.

B. Multi-Channel Aggregation

The graph neural network in CENTRIC is designed as a multi-hop neighbor diffusion model with a depth of Y.

1) Social Information Aggregation: For the user node u, p_u^S is defined to represent the user feature vector in the social graph G^S. p_u^S aggregates the features of trusted friends of user u and updates to the next layer:

$$p_u^{S,l+1} = \text{LeakyRelu} \sum_{f \in T_u} \left(W_1^S p_f^{S,l} + W_2^S \left(p_u^{S,l} \odot t_{uf} p_f^{S,l} \right) \right), \tag{1}$$

where $\text{LeakyRelu}(\cdot)$ denotes the LeakyRelu nonlinear activation function, $\odot$ denotes the element-wise product, W_1^S and W_2^S are the weight matrices, T_u denotes the set of trusted friends of user u and l indicates the layer of the neural network. It can be seen that each user's social feature p_u^S is jointly represented by his own feature and the features of friends in the upper layer.

In addition, the strength of social relations between users is not equal. For example, social relations can be intuitively divided into unilateral trust and mutual trust. According to social relationship theory [8], [9], users who trust each other tend to have more similar preferences, so a weight parameter t_{uf} is used

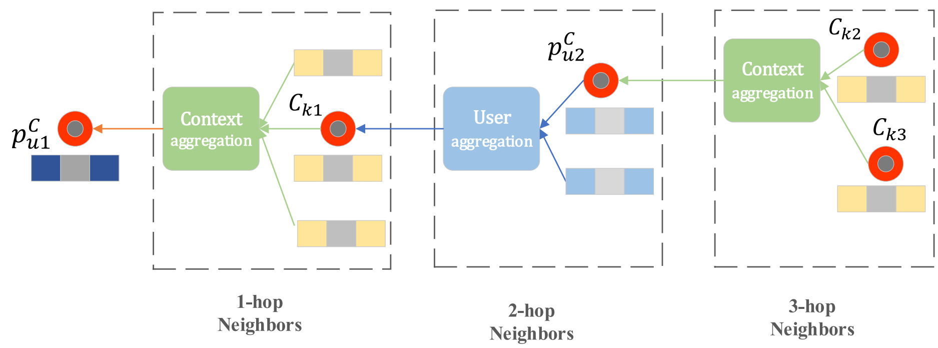


Fig. 2. User-context feature aggregation. User features and context features alternately aggregate the context/user information of their neighborhoods.

in (1) to adjust the influence of friends. t_{uf} is calculated as:

$$t_{uf} = \begin{cases} 1, u \in T_f \\ \frac{|T_u \cap T_f|}{\sqrt{|T_u| \cdot |T_f|}}, u \notin T_f \end{cases}, \tag{2}$$

where T_f denotes the trusted friends of user f. That is for mutual trust, the weight parameter is set to 1; for unilateral trust, the weight is set by measuring the number of common social friends.

2) User-Context Collaborative Information Aggregation: Current GNN-based social recommendation models ignore context information. In order to explicitly simulate the influence of context on user features, this study designs a user-context collaborative filtering module. By directly applying the GNN to the user-context bipartite graph, the contexts interacted by users are used to enhance user features and the users who interacted with contexts are used to enhance context features.

For the user node u, p_u^C is defined to represent the user feature vector in the user-context graph G^C. p_u^C aggregates the features of one-hop neighbor contexts and updates to the next layer:

$$p_u^{C,l+1} = \text{LeakyRelu} \sum_{k \in G^U(u)} \left(W_1^U c_k^l + W_2^U \left(p_u^{C,l} \odot c_k^l \right) \right). \tag{3}$$

Similarly, for the context node k, c_k is defined to represent the context feature vector in the user-context graph G^C. The update method of c_k is:

$$c_k^{l+1} = \text{LeakyRelu} \sum_{u \in G^C(k)} \left(W_1^C p_u^{C,l} + W_2^C \left(p_u^{C,l} \odot c_k^l \right) \right), \tag{4}$$

where $W_1^U, W_2^U, W_1^C, W_2^C$ are the weight matrices of the neural network, $G^U(u)$ denotes the set of contexts interacting with user u, and $G^C(k)$ denotes the set of users interacting with context k. As shown in Fig. 2, the module will capture more user-context collaboration signals as the user features and context features are updated alternately.

3) Users Collaborative Information Aggregation: Except social trust and context, users with similar interests and preferences can also help enhance the feature of target user. Known user-item interaction records can be used to calculate user similarity and item similarity, which is the theoretical basis of early collaborative filtering recommendations. In addition, context-aware recommendation theory believes that users have different preferences in different context situations [24], so both the similarity of historical interaction records and the similarity of context interactions are considered in CENTRIC. That is, for a user u and user v, their similarity coefficient s_{uv} can be calculated as:

$$s_{uv} = \frac{1}{2} \left(\frac{|I(u) \bigcap I(v)|}{\sqrt{|I(u)| \cdot |I(v)|}} + \frac{|C_u \bigcap C_v|}{\sqrt{|C_u| \cdot |C_v|}} \right), \tag{5}$$

where $I(u)$ and $I(v)$ denote the interacted items of user u and user v, C_u and C_v denote the contexts when u and v interacted with items.

Let $\theta \in (0, 1)$ denote the preset threshold, and v is considered as a similar user of u when $s_{uv} > \theta$. For the user node u, p_u^R is defined to represent the user feature vector in the similar user graph S_U. p_u^R aggregates the features of similar users and updates to the next layer:

$$p_u^{R,l+1} = \text{LeakyRelu} \sum_{v \in S_U(u)} \left(W_1^R p_v^{R,l} + W_2^R \left(p_u^{R,l} \odot s_{uv} p_v^{R,l} \right) \right), \tag{6}$$

where W_1^R and W_2^R are the weight matrices of the neural network, and $S_U(u)$ denotes the set of similar friends of user u.

4) Items Collaborative Information Aggregation: As mentioned above, known user-item interaction records can be used to calculate item similarity, and item features can also be enhanced from similar items. Since items are less susceptible to context than users, only the item's interaction records are considered when calculating item similarity. For item i and item j, the similarity coefficient s_{ij} can be calculated as:

$$s_{ij} = \frac{|R_i \bigcap R_j|}{\sqrt{|R_i| \cdot |R_j|}}, \tag{7}$$

where R_i and R_j represent the users who interacted with item i and item j respectively.

For item node i, q_i is defined to represent the item feature vector in graph S_I. q_i aggregates the features of similar items and updates to the next layer:

$$q_i^{l+1} = \text{LeakyRelu} \sum_{j \in S_I(i)} \left(W_1^I q_j^l + W_2^I \left(q_i^l \odot s_{ij} q_j^l \right) \right), \tag{8}$$

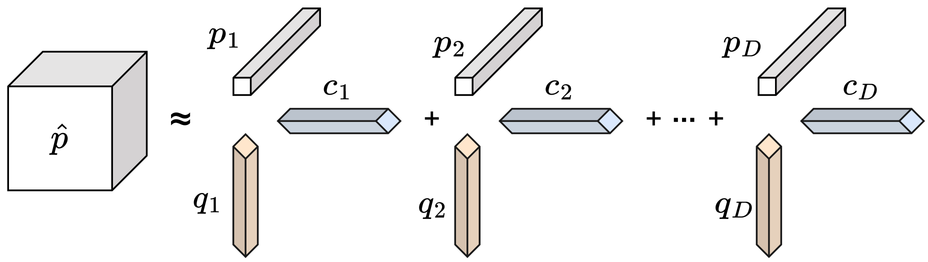


Fig. 3. Schematic diagram of CP decomposition. A third-order tensor is the sum of D rank-one tensors.

where $S_I(i)$ denotes the set of similar items, W_1^I and W_2^I are the weight matrices of the neural network.

C. Feature Fusion Based on Tensor Factorization

In CENTRIC, user features are enhanced by three sources: social friends, similar users and contexts. The user feature representation in each layer can be calculated as:

$$p_u^l = \alpha_1 p_u^{S,l} + \alpha_2 p_u^{R,l} + \alpha_3 p_u^{C,l}, \tag{9}$$

where $\alpha_1, \alpha_2, \alpha_3$ are hyperparameters, and $\alpha_1 + \alpha_2 + \alpha_3 = 1$, $\alpha_1 > 0$, $\alpha_2 > 0$, $\alpha_3 > 0$.

In addition, the representations of lower layers reflect more individual features, while the representations of upper layers reflect more neighbor features. To take advantage of the connections expressed by the output of different layers, we concatenate user, item and context features from all layers as the final node representation:

$$\begin{aligned} p_u &= \left[p_u^1, p_u^2, \ldots, p_u^Y\right], \\ q_i &= \left[q_i^1, q_i^2, \ldots, q_i^Y\right], \\ c_k &= \left[c_k^1, c_k^2, \ldots, c_k^Y\right]. \end{aligned} \tag{10}$$

Besides social information, recent studies [2], [12], [13] also considered the implicit influence of known interaction records in recommendation, which is called implicit feedback. Implicit feedback can help further alleviate data sparsity. As shown in the third part of Fig. 1, CENTRIC also utilizes implicit feedback to further enhance the feature representation of users and items. For the specified user u and item i, the implicit influence of observed users on target user is calculated as:

$$x_u^i = p_u + \max_{v \in R_U(i)} \{(p_u \odot p_v)\}. \tag{11}$$

Similarly, this study also considers the influence of observed items on target item:

$$y_i^u = q_i + \max_{j \in R_I(u)} \{(q_i \odot q_j)\}, \tag{12}$$

where x_u^i and y_i^u denote the user feature and item feature incorporated implicit feedback, $R_U(i)$ denotes the set of users who interacted with item i, and $R_I(u)$ denotes the set of items that have been interacted with by the user u. Here the max operation is used to select the most salient implicit feedback, which we found experimentally to be better than mean-pooling or summation.

Due to the added context, traditional matrix factorization is no longer applicable to feature fusion, because it can only handle the interaction of two kinds of features. As a powerful tool for processing higher-order data, tensor factorization can effectively capture user-item-context interaction features, so a feature fusion method based on CP decomposition is designed at the output layer. As shown in Fig. 3, CP decomposition [31] is a simple and fast tensor factorization model, which decomposes a higher-order tensor into the sum of rank-one tensors. The feature fusion method based on CP decomposition is expressed as:

$$\text{CP}\left(p_u, q_i, c_k\right) = \sum_{d=1}^{Y \times D} p_{ud} q_{id} c_{kd}, \tag{13}$$

where p_u denotes the feature vector of user node u, q_i denotes the feature vector of item node i, c_k denotes the feature vector of context node k, and $Y \times D$ is the embedding dimension after concatenation in (10).

Recent studies [22], [23] found that comprehensively considering feature representations that incorporate different information in feature fusion can help improve the robustness of the model. This is because the output is influenced by the combination of multiple features, and will not fluctuate greatly with the change of a single feature. So both feature representations with and without implicit feedback are considered in this study:

$$\begin{aligned} \hat{p}_{uik}^1 &= CP_1\left(p_u, q_i, c_k\right), \\ \hat{p}_{uik}^2 &= CP_2\left(x_u^i, y_i^u, c_k\right), \\ \hat{p}_{uik}^3 &= CP_3\left(x_u^i, q_i, c_k\right), \\ \hat{p}_{uik}^4 &= CP_4\left(p_u, y_i^u, c_k\right). \end{aligned} \tag{14}$$

Finally, the probability $\hat{p}_{uik}$ that user u selects item i under context k is denoted as:

$$\hat{p}_{uik} = \beta_1 \hat{p}_{uik}^1 + \beta_2 \hat{p}_{uik}^2 + \beta_3 \hat{p}_{uik}^3 + \beta_4 \hat{p}_{uik}^4, \tag{15}$$

where $\beta_1, \beta_2, \beta_3, \beta_4$ are hyperparameters, and $\beta_1 + \beta_2 + \beta_3 + \beta_4 = 1$, $\beta_1 > 0$, $\beta_2 > 0$, $\beta_3 > 0$, $\beta_4 > 0$.

D. Model Training

The purpose of Top-N recommendation is to select several items with the highest probability of interaction for users. In order to facilitate model training, the highest $\hat{p}_{uik}$ in all contexts is selected as the final predicted probability $\hat{p}_{ui}$ of user u for item i:

$$\hat{p}_{ui} = \max\left(\hat{p}_{ui1}, \hat{p}_{ui2}, \ldots, \hat{p}_{uiK}\right). \tag{16}$$

A widely used objective optimization function in model training is the cross-entropy loss:

$$L = -\sum_{(u,i)} \left[y_{ui} \log\left(\hat{p}_{ui}\right) + (1 - y_{ui}) \log\left(1 - \hat{p}_{ui}\right)\right], \tag{17}$$

where y_{ui} is 1 or 0 indicating whether there is a real interaction record. The optimizer of the model is mini-batch Adam. Since each user has a large number of unobserved items, 10 unobserved items are sampled as pseudo-negative feedback in each iteration, and positive feedback consists of known interaction records. We also adopt the dropout strategy [23], [36] to alleviate the overfitting in optimizing deep neural network models. By continuously minimizing the loss function L, the recommendation accuracy of the model will gradually improve.

E. Complexity Discussion

Space complexity: The model parameters of CENTRIC are composed of embeddings and weight matrices, so the space complexity of CENTRIC is $[P, Q, C] + [W_1^S, W_2^S, W_1^R, W_2^R, W_1^U, W_2^U, W_1^C, W_2^C, W_1^I, W_2^I]$. CP decomposition part requires $[P, Q, C]$ parameters, which is the same as the number of parameters needed for embeddings, so it does not add additional space complexity. For comparison, the space complexity of Diffnet++ [22] is $[P, Q] + [W_1, W_2, MLP_{1,2,3}]$ and that of DICER [23] is $[P, Q] + [W_1^S, W_2^S, W_1^R, W_2^R, W_1^I, W_2^I, MLP_{1,2,3}]$. The model proposed in this paper has an additional context embedding and more weight matrices. In fact, the weight matrices are shared among all users, items, and contexts, so the space complexity of CENTRIC is still acceptable.

Time complexity: Given M users, N items and K contexts, assume that each node directly connects L_S friends, L_R similar users, L_C contexts, L_I items and L_U users on average. Considering the network depth Y and embedding dimension D, the time complexity of the feature aggregation part in CENTRIC is: $O(Y(M(L_S + L_R + L_C)D + NL_ID + KL_UD))$. For comparison, the time complexity of Diffnet++ is $O(Y(M(L_S + L_I)D + NL_UD))$ and that of DICER is $O(Y(M(L_S + L_R)D + NL_ID))$. The time complexity of the CP decomposition part is: $O(MNK)$. The increased computational complexity mainly comes from the additional context information. However, the number of contexts K in real datasets is often very small, and $L_S, L_U, L_R, L_I \ll \min\{M, N\}$, so the time complexity of CENTRIC does not increase significantly compared with other GNN-based social recommendation models.

IV. Experimental Setups

This section first puts forward three research questions that need to be explored in experiments, then introduces the datasets and the preprocessing of context information. The evaluation metrics and experimental parameters are also given. Finally, various state-of-the-art baselines related to our work are introduced.

TABLE II
Statistics of the Datasets

Datasets	Ciao-DVD	LastFM	Epinions
Users	17,615	1,824	22164
Items	16,121	6,854	296277
Records	72,665	20,664	922267
Trusts	40,133	25,305	355813
Contexts	timestamp	timestamp	timestamp
Records Density	0.026%	0.165%	0.014%

A. Research Questions

We seek to answer the following questions in experiments:

(RQ1) What is the performance of CENTRIC compared to other methods? Does it outperform the state-of-the-art methods in terms of HR and NDCG on all datasets? (See †V-A)

(RQ2) What is the effect of the different components of CENTRIC on the performance? Do context, implicit feedback and tensor factorization help improve recommendation accuracy? (See †V-B)

(RQ3) How does the hyperparameters affect the performance of CENTRIC? How do the depths and embedding dimensions of GNN affect CENTRIC? (See †V-C)

B. Datasets

We conduct experiments on the Ciao-DVD[1], LastFM[2] and Epinions[3] datasets. Both contain social information and timestamps that can be used as context and are widely used in recommendation researches [12], [23]. Since only some records in the LastFM dataset are marked with timestamps, we remove the records without timestamps and renumber the user id and item id. The statistics of the datasets are shown in Table II.

We convert the timestamps in the LastFM dataset into specific times and divide them into five time periods of the day, thereby obtaining five contexts: morning (6:00am–11:00am), noon (11:00am–14:00pm), afternoon (14:00pm–18:00pm), evening (18:00pm–0.00am), and late night (0:00am–6:00am). The interactive records in the Ciao and Epinions are all from the same time period, so we convert the timestamps to seven days in a week as seven temporal contexts.

C. Evaluation Metrics

Hit Ratio (HR) [23] and Normalized Discounted Cumulative Gain (NDCG) commonly used in Top-N recommendations are selected as evaluation metrics for all experiments. HR indicates whether the recommended items are actually clicked by the user, while NDCG is more concerned about the ranking of hit items in the list, and the top-ranked items have higher calculation weights. The metrics are calculated as follows:

$$\mathrm{HR} = \frac{\sum_{i=1}^{N} hit(i)}{\min\left(N, |y_u^{\text{test}}|\right)}, \tag{18}$$

[1][Online]. Available: https://github.com/guoguibing/librec
[2][Online]. Available: https://files.grouplens.org/datasets/hetrec2011/
[3][Online]. Available: https://www.cse.msu.edu/\;tangjili/trust.html

$$\mathrm{NDCG} = \frac{\sum_{i=1}^{N} \frac{2^{hit(i)}-1}{\log_2(i+1)}}{IDCG}, \tag{19}$$

where N denotes the length of the recommendation list, $hit(i)$ is 1 or 0 to indicate whether the item was hit, $|y_u^{\text{test}}|$ denotes the number of interactive records of user u in the test set, $IDCG$ denotes the ideal sequence of items. Higher HR and NDCG represent higher recommendation accuracy.

D. Implementation Details

The datasets are divided into training set, validation set, and test set in a ratio of 7:1:2. The model first learns the main parameters on the training set, then the validation set is used to judge whether the model has been optimized to stability, and finally record the experimental results on the test set. To evaluate the performance, we randomly select 1000 unrated items as negative samples for each user. In order to reduce the occasionality of experiments, each experiment is repeated five times and the average value of evaluation metrics is recorded.

The main model parameters are set as follows: user similarity threshold $\theta_1 = 0.2$, item similarity threshold $\theta_2 = 0.2$, embedding dimension $D = 32$, depth of neural network is 3, learning rate is 0.001, batch size is 512, the slope of the negative part in LeakyRelu is set to 0.1. The hyperparameters $\alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{3}, \beta_1 = \beta_2 = \beta_3 = \beta_4 = \frac{1}{4}$, and we found that this setting can achieve optimal results through experiments. The dropout rate is set to 0.3 on Ciao and 0.5 on LastFM and Epinions. All experiments were conducted using an RTX 3090 graphics card.

E. Baselines

To assess the effectiveness of CENTRIC, we compare it with the following various state-of-the-art recommendation models. BPR and TrustSVD are traditional machine learning recommendation models, xDeepFM is a nueral FM model, NGCF, LightGCN, DiffNet, DiffNet++ and DICER are recent GNN-based recommendation models.

- **BPR** [32] is a classic pairwise sorting algorithm based on matrix factorization.
- **TrustSVD** [12] is an effective social recommendation algorithm, which integrates the user's social trust information on the basis of SVD++.
- **NGCF** [20] is a GNN-based recommendation model that captures high-order collaborative signals between user-item.
- **xDeepFM** [29] is a neural FM model which combines explicit and implicit high-order feature interactions.
- **LightGCN** [33] is a simplified variant of NGCF, which removes the inner product between neighboring nodes and transformation matrices to support more efficient recommendation.
- **DiffNet** [21] is a GNN-based social recommendation model that considers the transmission of indirect trust on social networks.
- **DiffNet++** [22] is an improved multi-channel model of DiffNet that further considers user-item collaborative signal.
- **DICER** [23] is a strong multi-channel baseline that uses both social information and user-item interaction history to enhance user and item feature representations.

V. Results and Analyses

In this section, we conduct three sets of experiments on three datasets to explore the research questions listed in †IV-A.

A. Overall Performance (RQ1)

First, a set of comparative experiments are conducted to observe the recommendation accuracy of CENTRIC and the above eight baselines. The lengths of the recommendation list are set to 5, 10 and 15 respectively. The experimental results are shown in Table III.

By analyzing the experimental results, it can be found that: as the most basic traditional recommendation model, BPR has much lower recommendation accuracy than other methods. Because BPR only considers the direct user-item interaction without incorporating any additional information, data sparsity causes the model to fail to accurately represent user and item features. As a powerful traditional recommendation model, TrustSVD utilizes both social trust and implicit feedback information, which effectively alleviates data sparsity, so the recommendation accuracy is significantly improved compared to BPR. NGCF is a deep learning model that utilizes GNN to model higher-order collaborative filtering information between users and items. Benefiting from the superiority of GNN for expressing structured data, NGCF outperforms traditional recommendation methods by a large margin. xDeepFM designs a compressed interaction network to explicitly learn higher-order feature interactions, so as to achieve better feature representation ability. LightGCN removes the transformation matrices and nonlinear activation function in NGCF, and combines the embeddings of all layers, which improves the recommendation accuracy and reduces the computational complexity. DiffNet simulates the diffusion of indirect trust signals on social networks, and DiffNet++ considers the user-item interaction graph on the basis of DiffNet, which further improves the recommendation accuracy. DICER constructs a relation-aware GNN model to process higher-order social relations and collaborative similarity relations, and utilizes both similarity information and social trust to enhance the feature representation of users and items, thus comprehensively performing the best among the baseline models.

The recommendation accuracy of our proposed CENTRIC is higher than that of DICER. Specifically, on the Ciao dataset, HR has an average increase of 5.7%, NDCG has an average increase of 6.9%; On the LastFM dataset, HR has an average increase of 4.7%, NDCG has an average increase of 6.5%; On the Epinions dataset, HR has an average increase of 2.8%, NDCG has an average increase of 3.2%. We attribute this mainly to user-context collaboration. By modeling multi-hop cooperative signals between users and contexts, the user's feature representation is simultaneously enhanced by social friends, similar users and contexts. Furthermore, considering implicit feedback can better alleviate data sparsity. In addition, we found in early

TABLE III
OVERALL PERFORMANCE. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD, AND THE SECOND BEST RESULT IS UNDERLINED. SIGNIFICANT IMPROVEMENTS OVER THE BEST BASELINE RESULTS ARE MARKED WITH * (T-TEST, $p < 0.05$)

Dataset	Models	HR@5	HR@10	HR@15	NDCG@5	NDCG@10	NDCG@15
Ciao	BPR	0.0836	0.1294	0.1325	0.0851	0.1129	0.1202
	TrustSVD	0.1272	0.1632	0.1683	0.107	0.1499	0.1543
	NGCF	0.163	0.1931	0.2562	0.152	0.1625	0.1874
	xDeepFM	0.1724	0.2136	0.2755	0.1614	0.1895	0.1967
	LightGCN	0.1951	0.2585	0.3277	0.1657	0.1878	0.2053
	DiffNet	0.1721	0.2153	0.2778	0.1601	0.1912	0.1974
	DiffNet++	0.1954	0.2663	0.3257	0.1672	0.1921	0.2032
	DICER	0.2224	0.3137	0.3759	0.1684	0.1972	0.2135
	CENTRIC	**0.2404***	**0.3292***	**0.392**	**0.1803***	**0.2106***	**0.2287***
	Improvement	8.09%	4.94%	4.28%	7.06%	6.79%	7.11%
LastFM	BPR	0.1254	0.163	0.1954	0.1117	0.1376	0.1502
	TrustSVD	0.1756	0.2102	0.2478	0.1562	0.1752	0.1841
	NGCF	0.2101	0.263	0.2862	0.1855	0.2121	0.2397
	xDeepFM	0.2232	0.2899	0.3347	0.1991	0.2367	0.2602
	LightGCN	0.2484	0.3195	0.3703	0.2229	0.2586	0.2837
	DiffNet	0.226	0.298	0.3456	0.2031	0.2431	0.2689
	DiffNet++	0.2461	0.3171	0.3652	0.2215	0.2574	0.2793
	DICER	0.2749	0.3475	0.4087	0.2475	0.2733	0.2954
	CENTRIC	**0.2889***	**0.3683***	**0.4211***	**0.265***	**0.2926***	**0.3114***
	Improvement	5.09%	5.98%	3.03%	7.07%	7.06%	5.41%
Epinions	BPR	0.1475	0.1521	0.1890	0.1464	0.1468	0.1601
	TrustSVD	0.1965	0.2348	0.2728	0.1931	0.2061	0.2202
	NGCF	0.2644	0.2985	0.3340	0.2681	0.2778	0.2902
	xDeepFM	0.2856	0.3343	0.3793	0.2845	0.3005	0.3167
	LightGCN	0.2981	0.3481	0.3938	0.2978	0.3136	0.3300
	DiffNet	0.2852	0.3376	0.3820	0.2832	0.3012	0.3172
	DiffNet++	0.3024	0.3501	0.3968	0.3022	0.3172	0.3338
	DICER	0.3125	0.3544	0.4025	0.3123	0.3258	0.3413
	CENTRIC	**0.3212***	**0.3664***	**0.4112***	**0.3231***	**0.3362***	**0.3518***
	Improvement	2.78%	3.39%	2.16%	3.46%	3.19%	3.08%

TABLE IV
PERFORMANCE COMPARISON WITH LESS TRAINING DATA

Dataset	Models	HR@5	HR@10	HR@15	NDCG@5	NDCG@10	NDCG@15
LastFM-316	DiffNet++	0.2256	0.2651	0.2914	0.2057	0.2336	0.2495
	DICER	0.2527	0.2757	0.3025	0.2442	0.2558	0.2583
	CENTRIC	**0.2821**	**0.3011**	**0.3278**	**0.2559**	**0.2745**	**0.2798**
LastFM-514	DiffNet++	0.2395	0.2862	0.3332	0.2198	0.2499	0.2705
	DICER	0.2722	0.3032	0.3448	0.2454	0.2713	0.2836
	CENTRIC	**0.2828**	**0.3291**	**0.3669**	**0.2546**	**0.2918**	**0.3034**
LastFM-712	DiffNet++	0.2461	0.3171	0.3652	0.2215	0.2574	0.2793
	DICER	0.2749	0.3475	0.4087	0.2475	0.2733	0.2954
	CENTRIC	**0.2889**	**0.3683**	**0.4211**	**0.265**	**0.2926**	**0.3114**

experiments that adding weight parameters when aggregating neighborhood information also played a minor role.

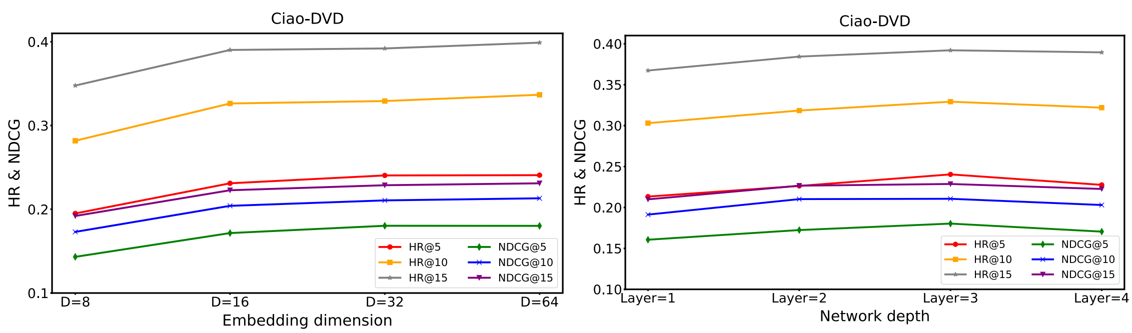
Finally, the recommendation performance of social recommendation algorithms when using few training data is also worth discussing. So, we run another set of experiments on LastFM dataset, setting the ratio of training set, validation set and test set to 5:1:4 and 3:1:6. The experimental results are shown in Table IV. It can be found that the recommendation accuracy of all models significantly decreases with the reduction of training data. However, compared with the two SOTA models DICER and DiffNet++, CENTRIC still has better recommendation performance. This indicates that considering social trust, context, and implicit feedback simultaneously helps to better alleviate data sparsity. The proposed model is still effective in the recommendation scenario with scarce training data.

B. Ablation Study (RQ2)

Compared with other GNN-based social recommendation models, the main contribution of CENTRIC is to enhance user representation with context information and implicit feedback, and utilize tensor factorization to fuse features at the output layer. Next, we observe the effects of context, implicit feedback and tensor factorization on recommendation accuracy through a set of ablation experiments. We design three variant models:

TABLE V
ABLATION STUDY

Dataset	Models	HR@5	HR@10	HR@15	NDCG@5	NDCG@10	NDCG@15
Ciao	CENTRIC-MF	0.2166	0.3097	0.3717	0.1604	0.1923	0.2103
	CENTRIC-IF	0.2307	0.3201	0.3844	0.1707	0.2014	0.2201
	CENTRIC-MLP	0.2397	**0.3294**	0.3897	0.1751	0.2084	0.2281
	CENTRIC	**0.2404**	0.3292	**0.392**	**0.1803**	**0.2106**	**0.2287**
LastFM	CENTRIC-MF	0.2695	0.3584	0.4063	0.2425	0.2739	0.2953
	CENTRIC-IF	0.2849	0.3593	0.4152	0.2554	0.2819	0.3025
	CENTRIC-MLP	0.2861	0.3642	0.4187	0.2603	0.2859	0.3102
	CENTRIC	**0.2889**	**0.3683**	**0.4211**	**0.265**	**0.2926**	**0.3114**
Epinions	CENTRIC-MF	0.3092	0.3573	0.4032	0.3077	0.3230	0.3392
	CENTRIC-IF	0.3140	0.3633	0.4081	0.3137	0.3294	0.3452
	CENTRIC-MLP	**0.3213**	**0.3674**	0.4109	0.3214	0.3352	0.3503
	CENTRIC	0.3212	0.3664	**0.4112**	**0.3231**	**0.3362**	**0.3518**



(a) Analysis of embedding dimensions.

(b) Analysis of network depths.

Fig. 4. Analysis of embedding dimensions and network depths on Ciao. (a) Analysis of embedding dimensions. (b) Analysis of network depths.

1) *CENTRIC-MF:* This variant removes the context-related part, only uses historical interaction records to calculate user similarity. User features are enhanced by similar users and social trust, and the output layer uses matrix factorization.
2) *CENTRIC-IF:* This variant removes the implicit feedback part, and the output layer only performs the probability prediction once: $\hat{p}_{uik} = CP(p_u, q_i, c_k)$.
3) *CENTRIC-MLP:* The output layer utilizes multi-layer perceptron (MLP) instead of tensor factorization to concatenate the features of users, items and contexts.

We compare these three variant models with CENTRIC on three datasets, and the experimental results are shown in Table V. By analyzing the experimental results, it can be found that context contributes the most in our model. After removing user-context collaboration module, the recommendation accuracy of CENTRIC-MF dropped significantly, which is close to the baseline DICER in †V-A. Specifically, HR dropped by 4.67% on average and NDCG dropped by 6.67% on three datasets. This result meets our expectations and visually demonstrates the effectiveness of utilizing context to enhance user features. In comparison, the contribution of implicit feedback is smaller, but it also steadily improves the accuracy of recommendation. Furthermore, tensor factorization also performs well in our model. Using MLP in the output layer does not significantly improve the recommendation accuracy. In contrast, CP decomposition has certain advantages in computational complexity. In fact, Rendle et al. proved in their work [34] that the MLP popular in neural networks is not always superior, and the feature fusion method based on dot product may be a better choice. Finally, combining user-context collaboration module, implicit feedback and tensor factorization can achieve the best results.

C. Analysis of Embedding Dimensions and Network Depths (RQ3)

In this subsection, we investigate the effect of hyperparameters on CENTRIC. The parameters θ_1 and θ_2 are used to control the number of similar users and similar items, which have relatively little effect on the model and will not be discussed in detail here. The embedding dimension D and depth of the neural network Y are two important parameters, which can greatly affect the recommendation performance of the model. Finally, two parameter tuning experiments are conducted. We got the same conclusion on the three datasets, here we only show the results on the Ciao dataset in Fig. 4.

By observing the experimental results, it can be found that as the size of the embedding dimension gradually increases from 8 to 16 and 32, the recommendation accuracy of CENTRIC increases steadily, which shows that a larger embedding dimension helps the embedding vector to capture more features. When the embedding dimension is increased to 64, the recommendation accuracy of the model has no obvious improvement. Since a larger embedding dimension will greatly increase the demand for computing resources such as video memory, we set the embedding dimension to 32 to balance the performance and computational efficiency of the model.

Furthermore, we observe similar results when adjusting the depth of the network. When the depth of the network is set to 1, the recommendation accuracy of the model is poor, because each node only considers the directly connected one-hop neighbors, which does not take advantage of the neural network to diffuse multi-hop neighbor information. As the depth increases, each node will incorporate more features of indirect neighbors, and the recommendation accuracy increases gradually. However, when the depth increases to 4, the recommendation accuracy actually declines. We analyze it because the model contains a lot of noise information, distant multi-hop neighbors are not enough to influence user preferences. Therefore, it is appropriate to set the depth of the neural network to 2 or 3 in our model, depending on the need to balance recommendation accuracy and computational efficiency in practical experiments. This observation is also consistent with some other recent GNN-based recommendation studies [21], [23], [35].

VI. Conclusion and Future Work

In this paper, we propose a new GNN-based social recommendation model with user-context collaboration and tensor factorization named CENTRIC. By designing a user-context collaborative filtering module, we explicitly utilize context information to further enhance the user's feature representation, and propose a tensor factorization fusion method at the output layer to fuse user-item-context interaction features. Multiple sets of experiments on three public datasets demonstrate that user-context collaboration and tensor factorization can help improve the accuracy of personalized recommendation, and the proposed model outperforms other state-of-the-art models. One limitation is that most current GNN-based social recommendation models have a huge number of parameters, and the introduction of context information will further increase the computational complexity of the neural network. Our next work plan is to investigate tensor network compression, which can be used to simplify the parameters of neural network and support faster recommendation. In addition, we also plan to collect datasets with richer data diversity or create a new dataset to observe the impact of different types of context information on recommendation.

References

[1] J. Chambua, Z. Niu, and Y. Zhu, "User preferences prediction approach based on embedded deep summaries," *Expert. Syst. Appl*, vol. 132, pp. 87–98, 2019.

[2] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[3] J. Chen et al., "Meta-path based neighbors for behavioral target generalization in sequential recommendation," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 3, pp. 1658–1667, May/Jun. 2022.

[4] G. Alexandridis, G. Siolas, and A. Stafylopatis, "Enhancing social collaborative filtering through the application of non-negative matrix factorization and exponential random graph models," *Data Mining Knowl. Discov.*, vol. 31, no. 4, pp. 1031–1059, 2017.

[5] Y. Liu, L. Chen, X. He, J. Peng, Z. Zheng, and J. Tang, "Modelling high-order social relations for item recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 9, pp. 4385–4397, Sep. 2022.

[6] J. Liu, C. Wu, and W. Liu, "Bayesian probabilistic matrix factorization with social relations and item contents for recommendation," *Decis. Support Syst.*, vol. 55, no. 3, pp. 838–850, 2013.

[7] B. Yang, Y. Lei, J. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1633–1647, Aug. 2017.

[8] P. V. Marsden and N. E. Friedkin, "Network studies of social influence," *Sociol. Methods Res.*, vol. 22, no. 1, pp. 127–151, 1993.

[9] S. Wasserman et al., "Social network analysis: Methods and applications," 1994.

[10] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proc. 17th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, 2008, pp. 931–940.

[11] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 287–296.

[12] G. Guo, J. Zhang, and N. Yorke-Smith, "TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 123–129.

[13] J. Zhao et al., "TrustTF: A tensor factorization model using user trust and implicit feedback for context-aware recommender systems," *Knowl. Based Syst.*, vol. 209, 2020, Art. no. 106434.

[14] Y. Zhang, C. Yang, K. Huang, and Y. Li, "Intrusion detection of industrial internet-of-things based on reconstructed graph neural networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 2894–2905, Sep.-Oct. 2023.

[15] Y. Zhu et al., "Recommending learning objects through attentive heterogeneous graph convolution and operation-aware neural network," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 4178–4189, Apr. 2023, doi: 10.1109/TKDE.2021.3125424.

[16] Y. Zhu, H. Lu, P. Qiu, K. Shi, J. Chambua, and Z. Niu, "Heterogeneous teaching evaluation network based offline course recommendation with graph learning and tensor factorization," *Neurocomputing*, vol. 415, pp. 84–95, 2020.

[17] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph neural networks in recommender systems: A survey," *ACM Comput. Surv.*, vol. 55, no. 5, pp. 1–37, 2022.

[18] F. Liu, Z. Cheng, L. Zhu, C. Liu, and L. Nie, "A2-GCN: An attribute-aware attentive GCN model for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 9, pp. 4077–4088, Sep. 2022.

[19] R. V. D. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," in *Proc. KDD Workshop Deep Learn. Day*, 2018.

[20] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 165–174.

[21] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang, "A neural influence diffusion model for social recommendation," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 235–244.

[22] L. Wu, J. Li, P. Sun, R. Hong, Y. Ge, and M. Wang, "DiffNet : A neural influence and interest diffusion network for social recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4753–4766, Oct. 2022.

[23] B. Fu, W. Zhang, G. Hu, X. Dai, S. Huang, and J. Chen, "Dual side deep context-aware modulation for social recommendation," in *Proc. World Wide Web Conf.*, 2021, pp. 2524–2534.

[24] G. Adomavicius and A. Tuzhilin, *Context-Aware Recommender Systems*. Boston, MA, USA: Springer, 2011.

[25] Y. Liu, S. Yang, Y. Xu, C. Miao, M. Wu, and J. Zhang, "Contextualized graph attention network for recommendation with item knowledge graph," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 181–195, Jan. 2023.

[26] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 79–86.

[27] L. Baltrunas and F. Ricci, "Experimental evaluation of context-dependent collaborative filtering using item splitting," *User Model. User-Adapted Interact.*, vol. 24, no. 1, pp. 7–34, 2014.

[28] W. Wu et al., "Improving performance of tensor-based context-aware recommenders using bias tensor factorization with context feature auto-encoding," *Knowl. Based Syst.*, vol. 128, pp. 71–77, 2017.

[29] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, “xDeepFM: Combining explicit and implicit feature interactions for recommender systems,” in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1754–1763.

[30] J. Wu et al., “Graph convolution machine for context-aware recommender system,” *Front. Comput. Sci.*, vol. 16, 2022, Art. no. 166614.

[31] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[32] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “BPR: Bayesian personalized ranking from implicit feedback,” in *Proc. 25th Uncertainty Artif. Intell.*, Montreal, QC, Canada, Jun. 18-21, 2009, pp. 452–461.

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

[34] S. Rendle, W. Krichene, L. Zhang, and J. Anderson, “Neural collaborative filtering vs. matrix factorization revisited,” in *Proc. 4th ACM Conf. Recommender Syst.*, 2020, pp. 240–248.

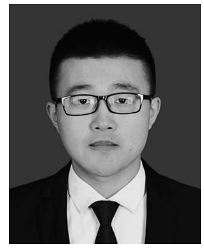
[35] J. Gong et al., “Attentional graph convolutional networks for knowledge concept recommendation in moocs in a heterogeneous view,” in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 79–88.

[36] P. Ren, Z. Chen, J. Li, Z. Ren, J. Ma, and M. Rijke, “RepeatNet: A repeat aware neural recommendation machine for session-based recommendation,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 4806–4813.

Wei Wang received the M.S. degree from the Shandong University of Science and Technology, Qingdao, China, in 2021. He is currently working toward the Ph.D. degree with the School of Information Science and Engineering, Shandong University, Qingdao, China.

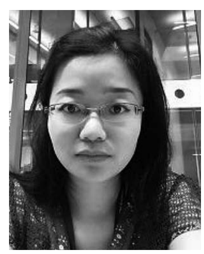
Zhenzhen Quan received the B.S. degree from the Shandong University of Science and Technology, Qingdao, China, in 2014 and the M.S. degree from Northeastern University, Shenyang, China, in 2017. She is currently working toward the Ph.D. degree with the School of Information Science and Engineering, Shandong University, Qingdao, China.

Siwen Zhao received the B.S. degree in 2019 from Shandong University, Qingdao, China, where she is currently working toward the Ph.D. degree with the School of Information Science and Engineering.

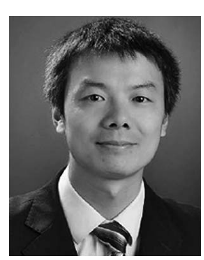


Guoqiang Sun received the M.S. degree from the Shandong University of Science and Technology, Qingdao, China, in 2020. He is currently working toward the Ph.D. degree with the School of Information Science and Engineering, Shandong University, Qingdao, China.

Yujun Li (Member IEEE) received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2001. His research interests include Big Data and Artificial Intelligence. Dr. Li is currently the Deputy Dean of the School of Information Science and Engineering, Shandong University, Qingdao, China, and the Chief Scientist or the Deputy Director of the Key Laboratory of IoT of the Ministry of Justice.



Xianye Ben (Senior Member IEEE) received the Ph.D. degree in pattern recognition and intelligent system from the College of Automation, Harbin Engineering University, Harbin, China, in 2010. She is currently a Full Professor with the School of Information Science and Engineering, Shandong University, Qingdao, China. She was enrolled by the Qilu Young Scholars Program of Shandong University. She has authored or coauthored more than 100 papers in major journals and conferences, such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON MULTIMEDIA, *Pattern Recognition*, and CVPR. Her current research interests include pattern recognition and image processing. Dr. Ben was the recipient of the Excellent Doctorial Dissertation by Harbin Engineering University.



Jianli Zhao received the Ph.D. degree from Northeastern University, Shenyang, China, in 2006. He is currently the Deputy Dean and Doctoral Supervisor with the School of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China. His research interests include tensor network, pervasive computing, and personalized recommendation.