Disentangled Modeling of Social Homophily and Influence for Social Recommendation

Nian Li, Chen Gao, Depeng Jin, *Member, IEEE*, and Qingmin Liao, *Senior Member, IEEE*

Abstract—Social recommendation leverages social information to alleviate data sparsity and cold-start issues of collaborative filtering (CF) methods. Most existing works model user interests following the assumption of *social homophily* based on social-relation data. The explicit modeling of *social influence*, which also largely affects user behaviors, has not been well explored. Considering user behaviors may be driven by social factors in today's information services (e.g., purchasing products shared by close friends on social e-commerce applications), these methods will be suboptimal. In this work, we propose a method modeling both social homophily-aware user interests and social influence as two essential effects on user behaviors for social recommendation, named as DISGCN (short for **DIS**entangled modeling of Social homophily and influence with **G**raph **C**onvolutional **N**etwork). Specifically, we devise a disentangled embedding layer to encode these two effects. Furthermore, two tailored graph convolutional layers are developed to disentangle them refinedly, leveraging the high-order embedding propagation in social-network graph from two aspects. Technically, first, the operation of attentive embedding propagation is adopted for capturing personalized social homophily-aware interests, and second, the item-gate-based embedding propagation is proposed for capturing item-specific social influence. In addition, to ensure the disentanglement of social influence, we propose a contrastive learning framework that endows corresponding embeddings with explicit semantics. Extensive experiments on two real-world datasets demonstrate the effectiveness of our proposed model. Further studies also verify the rationality and necessity of our designs. We have released the datasets and codes at this link: https://github.com/tsinghua-fib-lab/DISGCN.

Index Terms—Social recommendation, social homophily and influence, disentangled modeling, graph convolutional networks

1 Introduction

Recommender system plays a crucial role in today's information systems, filtering useless things for users to address the information-overload issue [1], [2], [3], [4]. On the other hand, social networks engage in information diffusion and propagation among users more and more frequently. Specifically, users can *show* (e.g., share, forward or recommend) items, further exposed to their friends in the social network. In this situation, user behaviors are affected by friends under social effects in real-world platforms. As a result, besides traditional user preference learning, the modeling of social network's effects is becoming an urgent and essential problem for improving user experiences as well as platform profit. This motivates a popular and influential research problem, *social recommendation* [5], [6].

Existing works of social recommendation [7], [8], [9], [10], [11], [12], [13] are mainly based on a primary and rough assumption of *social homophily* [14], indicating that social-connected users tend to have similar interests and interact with similar items. With this assumption, various recommendation models are proposed [7], [9], [10], [11], ensuring that the distance between friends' latent representations encoding intrinsic interests is closer compared with that of strangers. On the other hand, with kinds of platforms introducing more and more social attributes, friends usually bring explicit influence on users when they are making decisions. In this situation, users' behaviors do not necessarily reflect their interests. For example, users may be invited by friends to view some products on e-commerce platforms or mentioned by friends to watch some videos in micro-video applications. Therefore, how the social network genuinely affects user behaviors should be considered from two perspectives, *social homophily*-aware interests and *social influence*. Social homophily [14] refers to the phenomenon that the assumption above does exist in real-world information platforms, which is the motivation of most existing studies. For example, a user may *like* the same tweets with his/her friends on Twitter. Social influence [15], [16] refers to the fact that a user's behaviors on social platforms can affect or be affected by his/her friends. For example, users may purchase products that are not in line with their tastes but shared by their friends on social e-commerce platforms. Moreover, Kwahk *et al.* [17] also points out that social media on e-commerce platforms will promote the propagation of social influence, which further affects users' willingness to visit platforms or purchase products, through a questionnaire survey. Despite its essentiality, however, *social influence* is

- *Nian Li and Qingmin Liao are with the Department of Electronic Engineering, Shenzhen International Graduate School, Tsinghua University, Beijing 100190, China. E-mail: linian21@mails.tsinghua.edu.cn, liaoqm@tsinghua.edu.cn.*
- *Chen Gao and Depeng Jin are with the Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. E-mail: chgao96@gmail.com, jindp@tsinghua.edu.cn.*

Manuscript received 4 September 2021; revised 1 March 2022; accepted 16 May 2022. Date of publication 12 July 2022; date of current version 1 May 2023.
This work was supported in part by The National Key Research and Development Program of China under Grant 2020YFA0711403 and in part by the National Natural Science Foundation of China under Grants 61972223, 61971267, and U1936217.
(Corresponding author: Chen Gao.)
Recommended for acceptance by S. S. Bhowmick.
Digital Object Identifier no. 10.1109/TKDE.2022.3185388

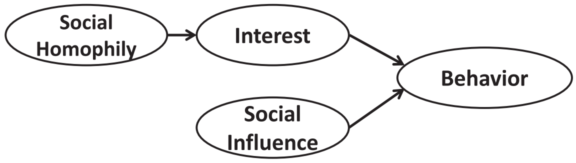


Fig. 1. Illustration of how social homophily-aware interests and social influence affect user behaviors.

neglected by existing works of social recommendation. Since interests do not entirely drive user behaviors when influenced by real-world social factors like co-purchasing with friends, the behaviors modeling with only preferences learning but without consideration of this effect will be suboptimal. Note that some works may have used the term *social influence* to express the interplay of user preferences [9], while in a more precise manner, it should be *social homophily*.

In this work, with more reasonable and complete consideration, we model both social homophily-aware interests and social influence for social recommendation. The former focuses on capturing user's intrinsic interests, and the latter handles the situations where social influence plays a crucial role in users' decision-making. To further distinguish these two factors, we model them in a disentanglement manner, which is demonstrated to be necessary for modeling user behaviors by many studies on social networks [16], [18], [19]. The affection mechanism of the two social factors on user behaviors is illustrated in Fig. 1.

Nevertheless, there are three non-negligible challenges as follows.

- *Social homophily and social influence take effects simultaneously.* As mentioned above, users' interaction behaviors with items are up to intrinsic interests as well as social influence from their friends. Thus, a set of entity (i.e., user and item) representations for modeling social homophily-aware interests in most existing works fails to handle all the possible scenarios of user behaviors.
- *Social homophily is personalized and social influence is item-specific.* Under the assumption of social homophily, even if users share common interests with friends, the commonalities vary among different friends. In other words, social homophily is not in a uniform or smoothing structure. Modeling such personalized and diverse social effects for accurate inference of user interests is essential. On the other hand, social influence takes effect not only dependent on users' friends, but also relevant to specific items or the way of showing items in a fine-grained manner. Here items indicate different things up to specific information services, such as commodities in social e-commerce applications or videos on micro-video platforms. In this situation, differentiating the item-specific social influence in the modeling of users' behaviors becomes vital but challenging. As for the way of showing items, we don't take this factor into account since it is hard to collect data in different ways in the real world, and we have only collected one-way sharing records between users and friends.
- *Social influence is always implicit and unseen.* In the setting of fundamental social recommendation problems, there is no explicit or labeled data for modeling social influence since both effects are always fused in observed user behaviors simultaneously. Therefore, from the perspective of representation learning, how to encode explicit semantics of the influence effects into user and item representations needs exploration.

To address these major challenges, we propose a novel model named DISGCN (short for **DIS**entangled modeling of **S**ocial homophily and influence with **G**raph **C**onvolutional **N**etwork) that disentangles the social homophily-aware interests and social influence. Specifically, we first devise a disentangled embedding layer that encodes these two effects separately. We then develop separate embedding propagation schemes based on graph convolution to further disentangle them in two sets of representations. Moreover, the schemes are tailored for their individual traits. Technically, we adopt attention mechanisms to capture the heterogeneity of social homophily. In terms of item-specific social influence, conventional information propagation that aggregates neighborhood embeddings can not differentiate which item the influence comes through. Thus we develop an item-gate-based embedding propagation scheme, i.e., to filter neighborhood influence with specific item representation as a controlling gate. Based on separate embedding propagation, social homophily-aware interests and social influence are further disentangled in corresponding representations. Besides, we propose contrastive learning framework as an auxiliary task to disentangle social influence, which endows corresponding representations with explicit meanings. Specifically, the discrimination between observed and fake social-influence behaviors is enlarged, and this helps to distill meaningful information and inject it into representations.

Then the contribution of this paper can be summarized as follows,

- We approach the research problem of social recommendation from a more accurate and complete perspective, which takes two most essential effects on user behaviors into consideration, i.e., social homophily-aware interests and social influence.
- We propose a novel method DISGCN, which models the effects of social homophily-aware interests and social influence simultaneously with disentangled representations. Then we develop separate embedding propagation mechanisms based on GCN to further disentangle these two effects as well as capture high-order information, which are also tailored for their individual traits, e.g., heterogenous and item-specific, respectively. Moreover, the contrastive learning framework is proposed to assists in disentangling the effect of social influence, which also endows corresponding representations with explicit meanings.
- We conduct extensive experiments on two real-world datasets. The experimental results show that our proposed method can achieve the best recommendation performance compared with state-of-the-art models. Further studies also verify the effectiveness of the disentangled representations, which are introduced for encoding the effect of social influence.

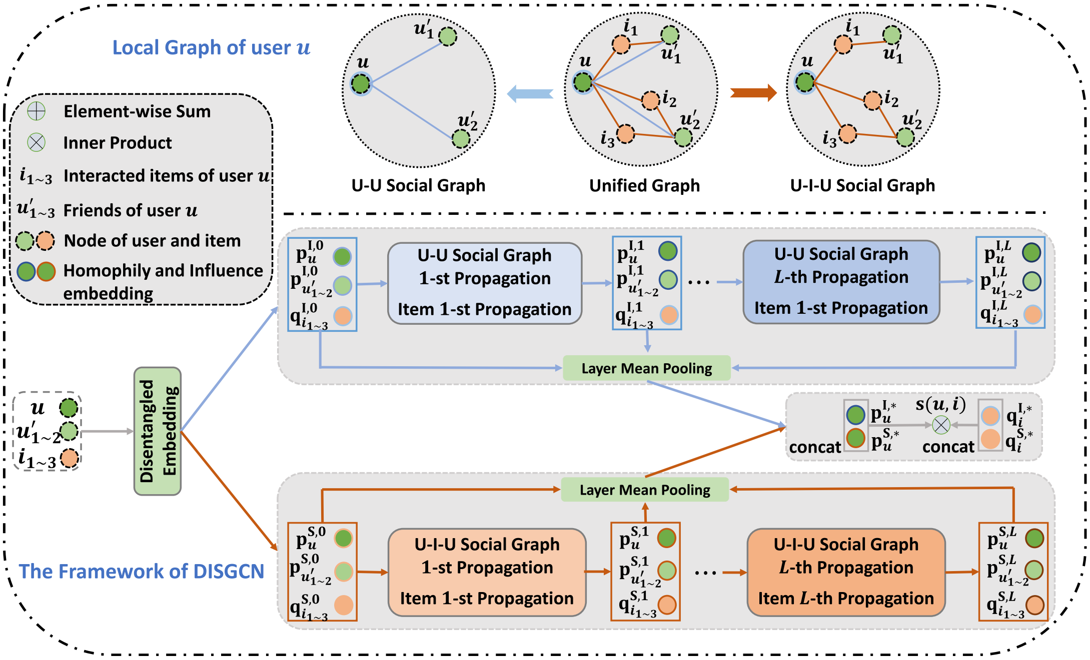


Fig. 2. The overall illustration of our DISGCN. The upper part shows local graph of the target user u, and u has two friends, u'_1 and u'_2. Additionally, u'_1 (u) brings explicit social influence on u (u'_1) through item i_1, and u'_2 (u) brings influence on u (u'_2) through item i_2 and i_3.

2 Problem Formulation

The task of social recommendation aims to leverage social-network data to enhance recommendation, of which there are two parts of data input. The first is the user-item interaction data (e.g., purchases and views), represented by a matrix Y, of which each entry is

$$y_{ui} = \begin{cases} 1, & \text{if user } u \text{ has interacted with item } i; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

In addition to the interactions, observed social-influenced behaviors serve as another necessary data input for modeling the social effect. For example, users can click/view/purchase products shared by friends on social e-commerce websites, such as Pinduoduo.[1] Another example is micro-video platforms, such as Wechat Video,[2] where users will be recommended and view videos liked by their friends following each other. These behaviors are mainly driven by social factors, rather than users' inherent preferences, which can be utilized as explicit and labeled data to model social influence. Formally, we define such social-influence behavioral data as $\mathcal{B} = \{(u, u,' i)|u' \textit{ shows } i \text{ to } u\}$,[3] where *shows* can indicate recommends, shares, or forwards, *etc*. With $\mathcal{B}$, we can obtain implicit social-relation data, represented by a matrix S, of which each entry is

$$s_{uu'} = \begin{cases} 1, & \exists (u, u,' i) \in \mathcal{B} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Then our problem can be formulated as follows.

1. https://pinduoduo.com
2. https://weixin.qq.com
3. Note that $\mathcal{B}$ will also be included in interaction data, which means that each record $(u, u,' i)$ leads to $y_{ui}, y_{u'i} = 1$.

Input. The interaction matrix Y, social-relation matrix S, and social-influenced behavioral data $\mathcal{B}$.

Output. A model that can estimate the likelihood that a given user u will interact with a given item i.

Then we can calculate the interaction likelihood of all the candidate items for each user, and construct the recommendation list with the top-ranked items.

3 Methodology

The framework of our proposed DISGCN is illustrated in Fig. 2, of which we take target user u as an example for a better understanding. The DISGCN model incorporates four components as follows.

- *Graph Construction:* We first construct a unified graph to represent interaction behaviors and social relations, with Y, S and $\mathcal{B}$ as data input. For modeling high-order homophily-aware user interests, we further construct user-user (U-U) social graph based on the unified graph. Similarly, user-item-user (U-I-U) social graph is constructed for modeling high-order social influence.
- *Disentangled Embedding Layer:* In order to model user interests and social influence separately in latent spaces, we assign two sets of embeddings for users/items. With disentangled low-dimension vectors, the two essential factors affecting user behaviors are both well captured.
- *Graph Convolutional Layer:* Considering the limitations of conventional GCN on the exploration of two effects with distinct traits, we devise separate graph convolution layers for tailored information extraction. Then the two sets of representations encoding high-order graph structure are disentangled

and contribute together to capturing user behaviors accurately.

- *Contrastive Learning:* To further disentangle the effect of social influence, we introduce the contrastive learning framework to distinguish whether influence exists in specific behaviors. Specifically, the observed social-influence behaviors are utilized as positive samples and randomly sampled fake behaviors are treated as negative samples, between which discrimination is encouraged. In this way, corresponding representations responsible for modeling social influence is endowed with explicit semantics.

3.1 Graph Construction

Since the data input of our problem is in a multi-relation structure, we first construct a unified graph to represent it overall. Specifically, the unified graph $G = (V, E)$ contains: 1) the node set V, which consists of user nodes $u \in \mathcal{U}$ and item nodes $i \in \mathcal{I}$, where $\mathcal{U}$ and $\mathcal{I}$ denote the set of users and items respectively; 2) the edges set E, which consists of three kinds of edges: interaction edge (u, i) with $y_{ui} = 1$, social-relation edge (u, u') with $s_{uu'} = 1$ and social-influence edge $(u, u^*, i^*)^4$ with $(u, u^*, i^*) \in \mathcal{B}$. Besides, in order to further disentangle the modeling of social homophily-aware interests and social influence, two kinds of social graph are extracted from the unified graph respectively, i.e., U-U and U-I-U graph, illustrated in the upper part of Fig. 2. Specifically, the U-U graph represents implicit social relations and the U-I-U graph exhibits item-specific social influence. The illustration denotes a demo dataset with $y_{ui_1}, y_{ui_2}, y_{ui_3} = 1, s_{uu'_1}, s_{uu'_2} = 1, \mathcal{B} = \{(u, u_1,' i_1), (u, u_2,' i_2), (u, u_2,' i_3)\}$. For a better understanding, we explain it in detailed. user u'_1 *shows* item i_1 to user u, and user u'_2 *shows* item i_2, i_3 to u. In addition, u'_1 and u'_2 are friends of user u. Besides, u'_1, u have interacted with i_1, and u'_2, u have interacted with i_2, i_3.

3.2 Disentangled Embedding Layer

As mentioned in the introduction, user behaviors can be affected by social homophily-aware interests and social influence. However, most current works focus on modeling user interests with only a set of embeddings for users and items. In contrast, we introduce a disentangled embedding layer to model these two factors simultaneously. Specifically, we construct two embedding matrices, $\mathbf{P^I} \in \mathbb{R}^{D\times|\mathcal{U}|}$ and $\mathbf{P^S} \in \mathbb{R}^{D\times|\mathcal{U}|}$, where D denotes the embedding size. Here we use the superscripts $\mathbf{I}$ and $\mathbf{S}$ to refer to user interests and social influence, respectively. To ensure the consistency in latent space, we assign embeddings for items corresponding to that of users, $\mathbf{Q^I} \in \mathbb{R}^{D\times|\mathcal{I}|}$ and $\mathbf{Q^S} \in \mathbb{R}^{D\times|\mathcal{I}|}$. In general, $\mathbf{P^I}$ is utilized for encoding information about what kinds of items the users will be interested in, and $\mathbf{Q^I}$ encodes information about what kinds of users the items' will be liked by. Basically, this is similar to most existing works of social recommendation. In terms of $\mathbf{P^S}$, it can be considered as a representation of how users will be influenced in social situations, and $\mathbf{Q^S}$ implies how items will

4. Here (u, u^*, i^*) represents a path of social influence $u^* \xrightarrow{i^*} u$. We use "edge" for unified expression, and it actually consists of edge (u^*, i) and (u, i).

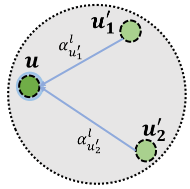


Fig. 3. Propagation on user-user social graph (U-U). Neighborhood information of different strengths is propagated to the node of the target user u.

bring influence, correspondingly. The embedding matrices convert the one-hot encoding of user u and item i as follows,

$$\mathbf{p}_u^{\mathbf{I}} = \mathbf{P^I}\text{one-hot}(u), \mathbf{p}_u^{\mathbf{S}} = \mathbf{P^S}\text{one-hot}(u),$$
$$\mathbf{q}_i^{\mathbf{I}} = \mathbf{Q^I}\text{one-hot}(i), \mathbf{q}_i^{\mathbf{S}} = \mathbf{Q^S}\text{one-hot}(i). \tag{3}$$

Note that these embedding matrices are the 0th layer's embeddings for embedding propagation, which will be introduced in Section 3.3.

3.3 Graph Convolutional Layers

To distill high-order social information, we adopt a GCN-based paradigm to perform further representation learning. As mentioned in the introduction, there are different traits in social homophily and influence. Specifically, social homophily is heterogeneous, and social influence always occurs with specific items. Thus we conduct separate embedding propagation on the constructed U-U and U-I-U social graph for disentangled modeling.

3.3.1 Embedding Propagation for Items

Since we only focus on the modeling of social factors (i.e., social homophily and influence) in this work, the embedding propagation for items simply follows most existing GNN-based CF methods for recommendation [20], [21]. Specifically, the propagation for the target item i is formulated as,

$$\mathbf{q}_i^{\mathbf{I},l+1} = \frac{1}{|\mathcal{U}_i|}\sum_{u\in\mathcal{U}_i}\mathbf{p}_u^{\mathbf{I},l}, \mathbf{q}_i^{\mathbf{S},l+1} = \frac{1}{|\mathcal{U}_i|}\sum_{u\in\mathcal{U}_i}\mathbf{p}_u^{\mathbf{S},l}, \tag{4}$$

where $\mathcal{U}_i$ denotes the set of users who have interacted with item i. The subscript l indicates the lth layer of propagation. The calculation of $\mathbf{p}_u^{\mathbf{I},l}$ and $\mathbf{p}_u^{\mathbf{S},l}$ will be introduced in Sections 3.3.2 and 3.3.3 respectively. In this simple but effective way, the model can capture CF signals from user-item relations for boosting recommendation.

3.3.2 Embedding Propagation on U-U Social Graph

Social homophily refers to that friends tend to have similar interests. With U-U social graph containing social connections between users, we build upon this social theory to perform embedding propagation. In addition, in real-world scenarios, the homophily is always heterogeneous, i.e., the common interests that users share vary among different friends. Considering this heterogeneity, we adopt an attention mechanism for propagation, shown in Fig. 3. Specifically, the procedure of embedding propagation for the target user u is formulated as follows,

$$\mathbf{p}_u^{\mathbf{I},l+1} = \sum_{u'\in\mathcal{S}_u}\alpha_{u'}^{l}\mathbf{p}_{u'}^{\mathbf{I},l}, \tag{5}$$

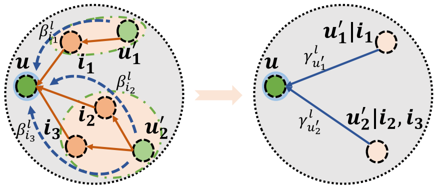


Fig. 4. Propagation on user-item-user social graph (U-I-U). Information from each friend of u is propagated and aggregated as individual social influence, with relevant items as controlling gates (left), then influence of different friends is aggregated as the overall influence for u (right).

where $\mathcal{S}_u$ denotes the neighbor set of u on U-U social graph, i.e., $\mathcal{S}_u = \{u' \mid s_{uu'} = 1\}$. We adopt the advice in [20] to remove non-linear activation function and feature transformation after aggregation for more effective and efficient representation learning, and similar operations are performed on other propagation layers of our method. The subscript l indicates the lth layer of propagation, and $\alpha^l_{u'}$ denotes the attention weight representing the homophily strength between user u' and target user u, calculated as follows,

$$\alpha^l_{u'} = \frac{e^{\hat{\alpha}^l_{u'}}}{\sum_{\hat{u}\in\mathcal{S}_u} e^{\hat{\alpha}^l_{\hat{u}}}}, \quad \hat{\alpha}^l_{u'} = \langle \mathbf{p}^{\mathbf{I},l}_u, \mathbf{p}^{\mathbf{I},l}_{u'} \rangle, \tag{6}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product. Note that we employ inner product to measure similarity between friends' embeddings for simplicity, and leave exploration of more complicated attention module as future work.

In short, with the embedding propagation of multiple layers, we obtain the representations modeling users' social homophily-aware interests, which can encode the high-order structure of U-U social graph.

3.3.3 Propagation on U-I-U Social Graph

Compared with the modeling of social homophily-aware interests, social influence has not been well explored in most existing works. Moreover, unlike social homophily that directly spreads among users with social connections, social influence always takes effect relevant to specific items. To be more specific, given a triplet (u_1, u_2, i), it can be regarded as a *social-influence path* and the item i is a *gate* that controls how two users, u_1 and u_2, influence each other. Taking social e-commerce as an example, the triplet (u_1, u_2, i) can represent that user u_2 shares the link of product i with user u_1, and invites u_1 to view and purchase it. Unfortunately, conventional information propagation can not distinguish which item social influence comes through. Thus we build upon this trait to propose an item-gate-based propagation scheme on the constructed U-I-U social graph. Technically, the propagation is performed in the following two stages, which is illustrated in Fig. 4. First, for a given user u, the social influence from a specific friend u' through multiple items, at the lth layer, is aggregated as follows,

$$\tilde{\mathbf{p}}^{\mathbf{S},l}_{u,u'} = \sum_{i\in\mathcal{S}_{u,u'}} \beta^l_i \Big(\mathbf{W}^l(\mathbf{p}^{\mathbf{S},l}_{u'} \odot \mathbf{q}^{\mathbf{S},l}_i)\Big) + \mathbf{p}^{\mathbf{S},l}_{u'}, \tag{7}$$

where $\mathcal{S}_{u,u'} = \{i \mid \exists(u, u,' i) \in \mathcal{B}\}$ denotes the set of items existing in social-influence behaviors between user u and u'. Here $\odot$ indicates element-wise product, $\mathbf{W}^l$ denotes the linear parameter matrix, and β^l_i is the diverse influential strength through different items. In general, β^l_i represents how strong the influence is between user u and u' through item i. To adaptively learn β^l_i, we propose to predict the influential strength in the contrastive learning framework, of which the details will be introduced in Eqn. (15) of Section 3.4. In the aggregation formula, $\mathbf{W}^l(\mathbf{p}^{\mathbf{S},l}_{u'} \odot \mathbf{q}^{\mathbf{S},l}_i)$ represents that, for each *social-influence path*, relevant item's representation is leveraged as a controlling gate. Generally, this gate aims to select what kind of social influence should be propagated. In terms of $\mathbf{p}^{\mathbf{S},l}_{u'}$, it is a skip-connection operation which can directly transfer the neighbor's information. To some extent, this operation can prevent the over-smoothing problem in graph convolution [22], [23].

After obtaining individual influence on the given user from all the friends, we aggregate them as,

$$\mathbf{p}^{\mathbf{S},l+1}_u = \sum_{u'\in\mathcal{S}_u} \gamma^l_{u'} \tilde{\mathbf{p}}^{\mathbf{S},l}_{u,u'}. \tag{8}$$

Besides, $\gamma^l_{u'}$ denotes the weight of social influence from friend u'. Specifically, the weight is obtained by an attention mechanism formulated as follows,

$$\gamma^l_{u'} = \frac{e^{\hat{\gamma}^l_{u'}}}{\sum_{\hat{u}\in\mathcal{S}_u} e^{\hat{\gamma}^l_{\hat{u}}}}, \quad \hat{\gamma}^l_{u'} = \langle \tilde{\mathbf{p}}^{\mathbf{S},l}_u, \tilde{\mathbf{p}}^{\mathbf{S},l}_{u,u'} \rangle, \tag{9}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product. Similar to Eqn. (6), we leave complicated attention module as future work.

In a nutshell, with item-gate-based information propagation for multiple layers, high-order connectivity on the U-I-U social graph containing social influence among users and items are encoded in corresponding representations.

3.3.4 Final Representations Combined

After L times of embedding propagation of graph convolutional layers, we obtain $L+1$ embeddings for users and items, including that of the raw 0th layer in Eqn. (3). Furthermore, all the layers of embeddings are aggregated with the mean-pooling operation, simple yet demonstrated effective, which is a widely-used manner in existing works [9], [24]. Then final embeddings are calculated as follows,

$$\begin{aligned}
\mathbf{p}^{\mathbf{I},*}_u &= \frac{1}{L+1}\sum_{l=0}^{L} \mathbf{p}^{\mathbf{I},l}_u, \mathbf{p}^{\mathbf{S},*}_u = \frac{1}{L+1}\sum_{l=0}^{L} \mathbf{p}^{\mathbf{S},l}_u, \\
\mathbf{q}^{\mathbf{I},*}_u &= \frac{1}{L+1}\sum_{l=0}^{L} \mathbf{q}^{\mathbf{I},l}_u, \mathbf{q}^{\mathbf{S},*}_u = \frac{1}{L+1}\sum_{l=0}^{L} \mathbf{q}^{\mathbf{S},l}_u.
\end{aligned} \tag{10}$$

3.4 Contrastive Learning

With all the layers of social-influence embeddings, we introduce endow them with explicit semantics further. Specifically, we treat the observed triplets in $\mathcal{B}$ as positive samples and randomly generate negative samples for them. The negative samples consist of two aspects, negative items and negative friends. The overall architecture of contrastive learning is illustrated in Fig. 5. In the illustration, the upper part represents the item aspect, which tends to learn why user u *shows* to his/her friend u' with item i rather than other items. Correspondingly, the friend aspect is shown in

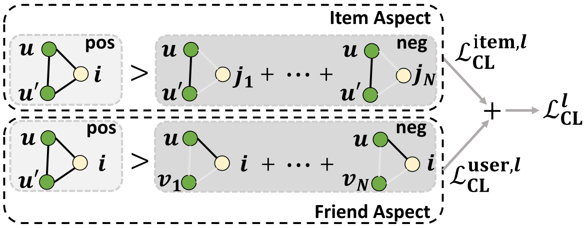


Fig. 5. Contrastive learning: Item-Aspect and Friend-Aspect.

the lower part, which tends to learn why user u *shows* item i to his/her friend u' rather than other users. The procedure of contrastive learning can be implemented in each layer, and then the objective function is formulated as follows,

$$\mathcal{L}_{\mathbf{CL}}^{\mathrm{item},l} = -\sum_{\mathcal{Q}_{u,u'}} \log \frac{\exp\big(g^l(u,u,'i)\big)}{\exp(g^l(u,u,'i)) + \sum_{n=1}^{N}\exp(g^l(u,u,'j_n))},$$
$$\mathcal{L}_{\mathbf{CL}}^{\mathrm{user},l} = -\sum_{\mathcal{Q}_{u,i}} \log \frac{\exp\big(g^l(u,u,'i)\big)}{\exp(g^l(u,u,'i)) + \sum_{n=1}^{N}\exp(g^l(u,v_n,i))}, \quad (11)$$

where $\mathcal{Q}_{u,u'}$ and $\mathcal{Q}_{u,i}$ are training sets for contrastive learning in item and friend aspect, respectively. Specifically, the training set consists of positive and negative samples as follows,

$$\mathcal{Q}_{u,u'} = \{(i, j_{1\sim N}) \mid (u,u,'i) \in \mathcal{B}, (u,u,'j_{1\sim N}) \notin \mathcal{B})\},$$
$$\mathcal{Q}_{u,i} = \{(u,'v_{1\sim N}) \mid (u,u,'i) \in \mathcal{B}, (u, v_{1\sim N}, i) \notin \mathcal{B})\}, \quad (12)$$

where N denotes the number of negative samples. Here we use j and v to denote the negative item and negative friend in the foregoing two negative-sampling aspects, respectively. For each sample, i.e., a triplet, we can predict whether it is an observed triplet or a fake one via a linear function,

$$g^l(u,u,'i) = \langle \mathbf{p}_u^{\mathbf{S},l}, \mathbf{p}_{u'}^{\mathbf{S},l}\rangle + \langle \mathbf{p}_u^{\mathbf{S},l}, \mathbf{q}_i^{\mathbf{S},l}\rangle + \langle \mathbf{p}_{u'}^{\mathbf{S},l}, \mathbf{q}_i^{\mathbf{S},l}\rangle, \quad (13)$$

of which there are three parts of matching scores. Generally, $\langle \mathbf{p}_u^{\mathbf{S},l}, \mathbf{p}_{u'}^{\mathbf{S},l}\rangle$ indicates whether user u has similar preferences with user u' and the degree of social influence between them; $\langle \mathbf{p}_u^{\mathbf{S},l}, \mathbf{q}_i^{\mathbf{S},l}\rangle$ indicates whether user u is interested in the item i shown from user u'; $\langle \mathbf{p}_{u'}^{\mathbf{S},l}, \mathbf{q}_i^{\mathbf{S},l}\rangle$ indicates whether user u' will interact with item i along with user u based on his/her intrinsic interests.

To this end, we take all the layers of embeddings into account, including item aspect and friend aspect, to obtain the overall loss function for contrastive learning as follows,

$$\mathcal{L}_{\mathbf{CL}} = \frac{1}{L+1}\sum_{l=0}^{L}\mathcal{L}_{\mathbf{CL}}^{\mathrm{item},l} + \frac{1}{L+1}\sum_{l=0}^{L}\mathcal{L}_{\mathbf{CL}}^{\mathrm{user},l}. \quad (14)$$

With optimizing the contrastive loss, social-influence representations are empowered with the ability to identify whether social influence exists or not among a triplet $(u,u,'i)$.

Note that β_i^l in Eqn. (7) represents the influential strength between two users via a specific item, which matches with the output of prediction function g. Therefore, we generate β_i^l based on g as follows,

$$\beta_i^l = \frac{e^{\hat{\beta}_i^l}}{\sum_{\hat{i}\in\mathcal{S}_{u,u'}} e^{\hat{\beta}_{\hat{i}}^l}}, \quad \hat{\beta}_i^l = g^l(u,u,'i). \quad (15)$$

This formula means that, the greater the probability of a triplet $(u,u,'i)$ being an observed social-influence behavior, the stronger information from u' to u through i is propagated.

3.5 Prediction and Model Optimization

For model prediction, we calculate the matching score between a user and item utilizing the widely-used and powerful inner product [20], [21], [24] of their final embeddings,

$$s(u,i) = s^{\mathbf{I}}(u,i) + s^{\mathbf{S}}(u,i),$$
$$s^{\mathbf{I}}(u,i) = \langle \mathbf{p}_u^{\mathbf{I},*}, \mathbf{q}_i^{\mathbf{I},*}\rangle, \;\; s^{\mathbf{S}}(u,i) = \langle \mathbf{p}_u^{\mathbf{S},*}, \mathbf{q}_i^{\mathbf{S},*}\rangle. \quad (16)$$

In the overall score, $s^{\mathbf{I}}(u,i)$ denotes how well user preferences match item attributes, focusing on the modeling of user interests, and $s^{\mathbf{S}}(u,i)$ denotes how possible a user will be influenced in social scenarios in terms of interaction with a specific item.

For the optimization of the recommendation model, we adopt widely-used BPR [25] loss function in recommender systems. It emphasizes that observed user-item interaction should be assigned with a higher matching score than unobserved one. The formulation is,

$$\mathcal{L}_{\mathbf{REC}} = \sum_{(u,i,j)\in\mathcal{O}} -\log\sigma(s(u,i) - s(u,j)) + \lambda\|\Theta\|_2, \quad (17)$$

where $\mathcal{O} = \{(u,i,j)|y_{ui}=1, y_{uj}=0\}$ is the training set of interaction data, and j denotes randomly selected negative samples. Θ denotes all the learnable parameters in the model, and λ is the L2 normalization coefficient; $\sigma(\cdot)$ is the sigmoid function, and $s(u,i)$ is the overall matching score between user u and item i demonstrated in Eqn. (16). By optimizing in two separate latent spaces rather than a single space in most existing works, the social effects on interaction behaviors, i.e., social homophily-aware interests and social influence, are both captured in disentangled representations.

Model Training. We adopt alternative training to optimize the main loss $\mathcal{L}_{\mathbf{REC}}$ and contrastive-learning loss $\mathcal{L}_{\mathbf{CL}}$. To be more specific, in each epoch, we first minimize $\mathcal{L}_{\mathbf{REC}}$ and update all the learnable parameters with $\mathcal{O}$ batch by batch, then minimize $\mathcal{L}_{\mathbf{CL}}$ and update parameters relevant to social-influence modeling with training data for contrastive learning, i.e., $\bigcup_{u,u,'i|u\in\mathcal{U},(u,u,'i)\in\mathcal{B}}(\mathcal{Q}_{u,u'} \bigcup \mathcal{Q}_{u,i})$.

Since many deep learning frameworks, such as Tensorflow, PyTorch, *etc.*, provide the function of automatic differentiation, we omit the explicit gradient calculations of the loss with respect to parameters.

Time Complexity Analysis. The main operations of DISGCN come from propagation layers, contrastive learning, and prediction. For embedding propagation on the U-U social graph of L layers, the computational complexity is $\mathcal{O}(|Y^+|DL)$, where $Y^+ = \{(u,i)|y_{ui}=1\}$ and $|Y^+|$ is the number of user-item interactions. For embedding propagation on the U-I-U social graph of L layers, the computational complexity is $\mathcal{O}(|\mathcal{B}|(D^2+D)L + |\mathcal{S}^{\mathbf{S},+}|DL)$, where $\mathcal{S}^{\mathbf{S},+} = \bigcup_{u\in\mathcal{U}}\mathcal{S}_u^{\mathbf{S}}$ and $|\mathcal{S}^{\mathbf{S},+}|$ denotes the number of all the friend pairs with social influence. For contrastive learning, the propagated embeddings are directly utilized for strength calculations of social-

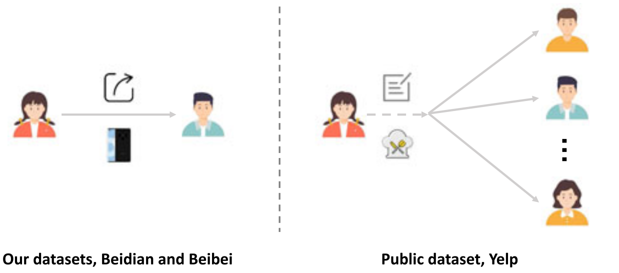


Fig. 6. A demo illustration to compare social influence on our datasets and public datasets.

TABLE 1
Dataset Statistics

Dataset	Beidian	Beibei
# Users	2,841	24,827
# Items	2,298	16,864
# Interactions	35,146	1,667,320
# Social relations	2,367	197,590
# Influence Behaviors	4,004	714,111

influence triplets, and the computational complexity is $\mathcal{O}(|\mathcal{B}|DL)$. Finally, embedding combination and prediction layer have computational complexity $\mathcal{O}(|Y^+|D)$ and $\mathcal{O}(|Y^+|DL)$, respectively. Combine all the computations above, the final complexity for each training epoch is $\mathcal{O}(|\mathcal{B}|D^2\ L + (|\mathcal{B}| + |\mathcal{S}^{\mathrm{S},+}| + |Y^+|)DL + |Y^+|D)$.

As for the training time of each epoch, all the GCN-based models cost similar seconds, including our DISGCN. Although our model needs more time to conduct negative sampling for contrastive learning, this can be solved via storing enough training samples in advance. In summary, the computation and time complexity of our DISGCN is comparable with GCN-based models, such as NGCF [21] and DiffNet [9].

4 Experiment

In this section, we conduct extensive experiments on two real-world datasets collected from social e-commerce platforms for evaluation of DISGCN. The following research questions will be answered:

- *RQ1:* Does DISGCN improve recommendation performance compared with state-of-the-art models?
- *RQ2:* How do specially modeled social-influence representations play a significant role in improving recommendation performance? Dose the embeddings are endowed with explicit semantics of social influence?
- *RQ3:* How do the hyper-parameters affect the effectiveness of our DISGCN?

4.1 Experimental Settings

4.1.1 Dataset

Since the explicit modeling of social influence is seldom considered in existing works of social recommendation [9], [26], there are no public datasets suitable. Although some public datasets contain users' public behaviors (e.g., retweets on Twitter or rating on Yelp), this reflects coarse-grained social influence a user brings to all the friends. However, the motivation of our work is to model fine-grained social influence a user brings to each friend. A simple illustration is shown in Fig. 6 to represent the difference.

Therefore, we collect two datasets with users' sharing behaviors from two social e-commerce platforms, Beidian[5] and Beibei.[6] On these two platforms, a user can choose to share some items with his/her friends. When a shared link from a user is clicked by his/her friend, we can collect a $\langle \mathrm{user, user, item} \rangle$ record, which is considered as a kind of social-influence interaction. In this social scenario, users may interact with items primarily driven by their friends' influence, in other words, their interests don't entirely lead to the behaviors. We summarize the statistics of the two datasets in Table 1. Although the scale of Beidian is relatively smaller, there is no other public dataset suitable. We release the datasets along with codes at this public link: https://github.com/tsinghua-fib-lab/DISGCN, which we believe will benefit the community.

- *Beidian.* This is a rising platform for social e-commerce and there are various kinds of products sold on the mobile App.
- *Beibei.* This is another social e-commerce service operated by the same company as Beidian does. Different from Beidian, Beibei focuses on Maternal and Child supplies sales.

Although operated by the same company, these two platforms are entirely independent. To be more specific, they target different users and sell different products. Thus they can be considered as two distinct datasets for the evaluation of recommendation models.

We randomly select 70% of historical interactions of each user as training data, 10% as validation data for tuning hyper-parameters, and 20% as test data for evaluation, which is a widely-used paradigm [27], [28].

4.1.2 Evaluation Metrics

We utilize two widely-used metrics, Recall and NDCG, for evaluation. Here we adopt the *full-ranking* manner, which is demonstrated by [29] to be more accurate and reliable compared with the sampled manner. Specifically, the metrics are defined as follows,

- *Recall@K* measures the ratio of test items that have been successfully recalled in the top-K recommendation list.
- *NDCG@K* assigns higher scores for the test items with higher rankings in the top-K recommendation list, which emphasizes that test items should be ranked as higher as possible.

Since the scales of the two datasets are significantly different (the number of items in the Beibei dataset is far larger), we set different ranges of K for them, to ensure stable results. Specifically, we set $K = 5, 10, 20$ for Beidian and $K = 20, 40, 80$ for Beibei.

5. https://www.beidian.com
6. https://www.beibei.com

4.1.3 Baselines

We compare our method with two categories of baselines, CF models and social recommendation models.

CF Models.

- *MF-BPR* [25] This is a state-of-the-art model based on matrix factorization(MF) in CF methods. It optimizes pairwise loss with the assumption that observed interactions should be assigned with higher matching scores than unobserved ones.
- *NGCF* [21] NGCF is the state-of-the-art graph neural network model that conducts embedding propagation on user-item bipartite graphs. The high-order information is captured in user/item representations.
- *LightGCN* [20] LightGCN is a simplified but more effective GNN model for recommendation compared with NGCF. This model only contains neighborhood aggregation for collaborative filtering, which is implemented as linear summation of embeddings.

Social Recommendation Models.

- *CSR* [30] This is a recent advanced model based on MF in social recommendation. The diverse social homophily is modeled with a characteristic regularization term. There are two variants with different optimization manners in the proposed method, and we adopt the static training manner.
- *EATNN* [26] This work develops a personalized transfer schema for user preferences modeling in both item and social domains with attention mechanism and designs an effective optimization method to train the whole model without negative sampling.
- *DiffNet* [9] This is a state-of-the-art social recommendation model based on GCN. An embedding propagation mechanism is proposed on social graph to simulate the diffusion process of social homophily.
- *GraphRec* [31] As a state-of-the-art method for social recommendation, GraphRec models user-user and user-item relations coherently with GCNs. In addition, attention mechanism is leveraged to model heterogeneous strengths in relations above. Since this model is proposed for the task of rating prediction rather than ranking, we compare it with our model individually.
- *DisenGCN* [32] This work introduces the disentangled graph convolutional network to learn disentangled node representations for different factors of connections on the graph. In this work, user-user social relations and user-item interactions make up a unified graph, on which we implement the method to encode the effects of user interests and social influence with node representations, setting the number of disentanglements as 2.
- *DICE* [33] DICE is a general framework to disentangle user interests and conformity for recommendation. Here conformity refers to users following other people, and can be considered as coarse-grained social influence from all the other users.

4.1.4 Hyper-Parameter Settings

Our DISGCN is implemented with Tensorflow. Following existing works [34], [35], [36], [37], the embedding size is fixed to 32 for our model DISGCN and EATNN, and 64 for other baselines. Note that both DISGCN and EATNN have two sets of embedding for users/items, thus a half embedding size is utilized for a fair comparison. Besides, user and item embeddings in the 0th layer $\mathbf{P^I}, \mathbf{P^S}, \mathbf{Q^I}, \mathbf{Q^S}$ are initialized as normal distribution $\mathcal{N}(0, 0.01)$ following [8], [9].

The batch size is fixed as 128 and 4,096 for Beidian and Beibei respectively, since their interaction records are in different scales. For the loss function $\mathcal{L}_{\mathbf{REC}}$ in Eqn. (17), the number of negative samples is fixed to 10 for an observed interaction (u, i). In terms of contrastive loss function $\mathcal{L}_{\mathbf{CL}}$, we also samples 10 negative items and friends for an observed influence-behavior $(u, u,' i)$ for comparable efficiency of training.

All the baseline methods' sampling number is also set as 10, except for EATNN, whose training strategy is in a non-sampling manner. In terms of other hyper-parameters, we apply a grid search to confirm the optimal settings: the learning rate is searched among [1e-3, 3e-4, 1e-4, 3e-5, 1e-5, 3e-6, 1e-6], and L2 normalization coefficient λ is tuned among [1e-2, 1e-3, 1e-4, 1e-5, 1e-6]. Besides, the layers of graph convolution are tuned among [1, 2, 3, 4] for all the GCN-based models. Moreover, we utilize the NDCG@20 on the validation data to choose suitable hyper-parameters. Following [21], for all the GCN-based baselines, we adopt the pretrain mechanism, i.e., leverage learned embeddings by MF-BPR [38] with L2 normalization as initialization of user and item embeddings. Since DisenGCN divides embedding into several (2 in our problem) chunks, it's not reasonable to pretrain with BPR, and we initialize embeddings randomly. All the experiments are conducted with a 12 G NVIDIA TITAN Xp GPU and 2.20 GHz Intel(R) Xeon (R) E5-2650 CPUs.

4.2 Overall Performance (RQ1)

We first compare the overall performance of our DISGCN with other baselines. The results on two datasets are reported in Tables 2 and 3, respectively. To make the results reliable, we run competitive experiments with different random seeds and then calculate the averaged performance. From the results, we have the following observations.

- *Our DISGCN achieves the best performance.* Our DISGCN outperforms all baseline methods consistently across all Recall and NDCG metrics. The improvement of our model compared with the best baseline is 9.36% $\sim$ 14.57% and 3.60% $\sim$ 4.48% on the Beidian and Beibei datasets respectively, which can be claimed as significant following existing works [39], [40]. Note that DiffNet, a GNN-based state-of-the-art model for social recommendation, does not necessarily obtain the best performance among baselines, especially for the Beibei dataset. Besides, as a simplification but improved version of NGCF, LightGCN can not always perform better. Therefore, the stable and consistent improvement further demonstrates the effectiveness of our model.
- *The modeling of social homophily-aware interests is significant for improving recommendation.* As mentioned in the introduction, most existing models of social recommendation focus on modeling user interests

TABLE 2
Overall Top-K Recommendation Performance on the Beidian Dataset

	Method	Recall@5	NDCG@5	Recall@10	NDCG@10	Recall@20	NDCG@20
CF-based	MF-BPR	0.0552	0.0501	0.0958	0.0657	0.1533	0.0850
	NGCF	0.0598	0.0531	0.1040	0.0703	0.1610	0.0895
	LightGCN	0.0525	0.0484	0.0908	0.0634	0.1491	0.0828
Social-based	CSR	0.0571	0.0519	0.0993	0.0678	0.1605	0.0882
	EATNN	0.0599	0.0536	0.0994	0.0688	0.1630	0.0901
	DiffNet	<u>0.0604</u>	<u>0.0541</u>	<u>0.1051</u>	<u>0.0712</u>	<u>0.1659</u>	<u>0.0916</u>
	DisenGCN	0.0429	0.0377	0.0761	0.0511	0.1264	0.0681
	DICE	0.0532	0.0489	0.0922	0.0638	0.1446	0.0813
Ours	**DISGCN**	**0.0692**	**0.0615**	**0.1149**	**0.0790**	**0.1841**	**0.1020**
	Improv.	14.57%	13.80%	9.36%	10.95%	11.01%	11.27%

TABLE 3
Overall Top-K Recommendation Performance on the Beibei Dataset

	Method	Recall@20	NDCG@20	Recall@40	NDCG@40	Recall@80	NDCG@80
CF-based	MF-BPR	0.1034	0.0966	0.1623	0.1136	0.2360	0.1362
	NGCF	0.1034	<u>0.0975</u>	0.1603	0.1137	0.2308	0.1352
	LightGCN	<u>0.1040</u>	0.0974	<u>0.1633</u>	<u>0.1144</u>	0.2362	0.1368
Social-based	CSR	0.1039	0.0973	0.1620	0.1139	0.2360	0.1369
	EATNN	0.1006	0.0943	0.1535	0.1092	0.2216	0.1300
	DiffNet	0.1035	0.0975	0.1624	0.1142	<u>0.2376</u>	<u>0.1371</u>
	DisenGCN	0.0853	0.0813	0.1357	0.0958	0.1996	0.1156
	DICE	0.0606	0.0546	0.1025	0.0679	0.1547	0.0838
Ours	**DISGCN**	**0.1087**	**0.1012**	**0.1694**	**0.1185**	**0.2479**	**0.1424**
	Improv.	4.48%	3.81%	3.68%	3.60%	4.34%	3.89%

based on a basic assumption of social homophily. Specifically, the distance between social-connected users in latent space is shortened, and this will be helpful to improve recommendation performance as well as capture user interests more accurately. For instance, CSR and DiffNet outperform classical MF-BPR on the two datasets, demonstrating that social information does take effect. Moreover, although GCN-based models have a stronger ability of representation learning, CSR can achieve better performance compared with NGCF on most metrics on the Beibei dataset. This comparison result further addresses the necessity of capturing social homophily. Additionally, thanks to the ability of interest-factor disentanglement from interaction data with social influence, our DISGCN will model users' real and intrinsic preferences better. Therefore, DISGCN obtains relatively greater improvements when K is smaller.[7]

- *The modeling of social influence is especially essential when the interaction behaviors contain this effect.* As mentioned in Section 2, the interaction data contains some users' social-influence behaviors, i.e., sharing of product links. In other words, these interactions can not be modeled adequately with only considering users' intrinsic interests, on which most existing works concentrate. Therefore, DiffNet only obtains slight performance improvement compared with MF-BPR. On the other hand, DICE performs badly, although it models coarse-grained social influence. In contrast, our DISGCN can disentangle the effect of social influence and performs best consistently. This comparison demonstrates that modeling fine-grained social influence is essential for real-world and complicated user-item interactions in social scenarios.
- *Conventional information propagation is not capable of modeling both user interests and social influence simultaneously.* DisenGCN, a disentangled GCN model that attempts to capture independent factors (user interests and social influence in our problem) of complex interactions on graphs with disentangled node representations, performs worst on the two datasets. Its failure in modeling user behaviors can be explained in the following two aspects. On the one hand, the same information propagation on the unified graph can not distinguish these two factors. On the other hand, the conventional propagation scheme is not capable of capturing item-specific social influence. By contrast, our DISGCN disentangles social homophily-aware user interests and social influence with attentive and item-gate-based propagation, respectively. Moreover, item-specific social influence can be differentiated well since relevant items are leveraged as gates to control information propagation. As a result, DISGCN captures user behaviors accurately

7. Metrics of smaller top-Ks mean more strict judgment on the recommendation list.

TABLE 4
Overall Top-K Recommendation Performance in a Sampling Manner on the Beidian and Beibei Datasets

Dataset	Method	Recall@1	Recall@3	NDCG@3
Beidian	GraphRec	0.1133	0.2552	0.1947
	DISGCN	0.9416	0.9965	0.9752
Beibei	GraphRec	0.1917	0.3843	0.3007
	DISGCN	0.5032	0.7944	0.6732

with disentangled representations and consistently obtains the best performance.

As mentioned above, GraphRec is proposed for the task of rating prediction [41], [42], thus we train the model with negative sampling for adapting to the ranking task. Specifically, observed interactions are directly treated as rating 1, and randomly sampled negative items are rating 0. In terms of evaluation, since running with official implementation[8] for all the user-item pairs is very slow, we adopt a widely-used sampling manner [8], [9], i.e., randomly select one item in the original test set and other 99 negative items to rank these 100 items. The comparison results are shown in Table 4, and our model performs significantly better than GraphRec. It can be explained that GraphRec is more suitable for rating-prediction tasks as it is designed to be (a solution of the rating-prediction-based social recommendation problem).

In summary, our DISGCN obtains steady performance improvement compared with state-of-the-art recommendation models. Moreover, the modeling of social homophily-aware interest and social influence are both essential for accurate inference of user behaviors. Besides, conventional embedding propagation that directly collects neighborhood information without differentiating where influence comes from is not suitable to model item-specific social influence. In contrast, our proposed item-gate-based propagation scheme can handle this refinedly.

4.3 The Effectiveness of Social-Influence Representations (RQ2)

In this section, we further investigate learned social-influence embeddings from two aspects. On one hand, we investigate users' recommendation performance more refinedly on the Beibei dataset to evaluate the critical effects of influence embeddings for improving recommendation. On the other hand, detailed analysis of embeddings is conducted to demonstrate the semantics of embeddings about social influence.

4.3.1 Effects of Improving Recommendation

We first distinguish users based on the strength of social influence. Specifically, only interaction records in the testing data that coexist with certain training records in original influence-behavior data $\mathcal{B}$ are reserved for evaluation. In this way, the testing items of each user are all influential ones *shown* (e.g., shared or recommended) by friends, that is to say, they bring social influence explicitly to users. With these influential testing records, we then divide users into several groups according to the number of testing items, which indicates the degree of how users are influenced when making decisions. Furthermore, we design two kinds of evaluation manners,

- *1) Half embeddings:* Evaluate utilizing only interests embeddings for our DISGCN. For fair comparison, the first half dimensions of embeddings is utilized for LightGCN, DiffNet and CSR.
- *2) Full embeddings:* Evaluate utilizing both interests and influence embeddings (i.e., concatenation) for our DISGCN and full embeddings for LightGCN, DiffNet and CSR.

Under the comparison of these two evaluation manners, whether disentangled influence embeddings have learned information of social influence will be verified. After dividing users into groups with varying degrees of social influence, we calculate the average metrics for each group and display the results in Fig. 7. We set suitable thresholds in dividing groups to ensure there are enough users for each group and avoid bias. There are two observations from the comparison results:

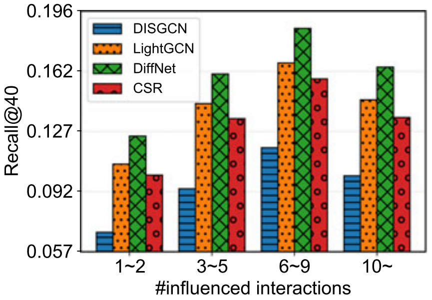
- The *NDCG* metric obtains higher values if users have more interactions. We attribute this improvement to the more accurate preference modeling, that is to say, the items that users are intrinsically interested in are ranked higher in the recommendation list. However, *Recall* drops significantly when users have too many interactions influenced by friends, i.e., in the user group "$>$ **10**". This demonstrates that user behaviors under the strong social influence vary greatly, making accurate recommendations with only preference learning difficult. In summary, it's essential to take social influence into consideration explicitly since users' decisions are driven by not only their interests but also social factors.
- In the evaluation 1), DISGCN performs badly across each user group with different influential testing records. However, it obtains significant performance improvement in the evaluation 2). This shows that influence embeddings play a critical role in improving recommendation. As a matter of fact, the representations have been endowed with rich information of social influence. Thus they are much helpful in capturing behaviors of users who are more inclined to be influenced by friends.

4.3.2 Semantics of Social-Influence Embeddings

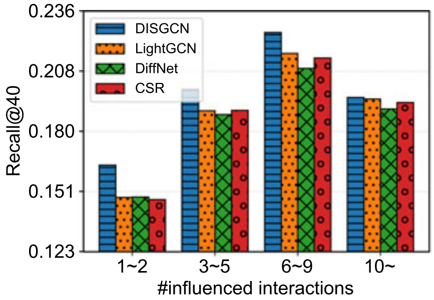
We first investigate semantics by studying embedding distance. We present the distribution of user-friend embedding distance and user-stranger embedding distance in Fig. 8. As we can see, the latter is significantly skewed towards larger values than the former. In other words, friends with social influence are closer to users in the embedding space compared with strangers without social influence. This is exactly what contrastive learning (CL) aims to learn, i.e., assists to distinguish friends from strangers for each user. Therefore, CL endows embeddings with this semantics.

Moreover, we also visualize embeddings for intuitive understanding. We select two user groups, according to the strength of user influence. Specifically, the first group of

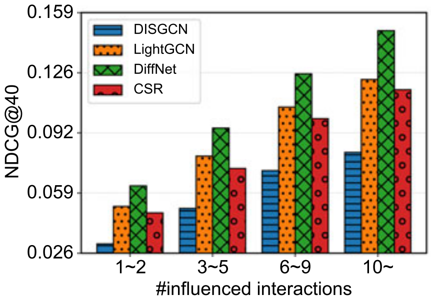
8. https://github.com/wenqifan03/GraphRec-WWW19



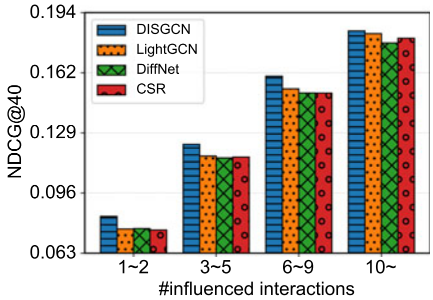
(a) Recall@40, half embeddings



(b) Recall@40, full embeddings



(c) NDCG@40, half embeddings



(d) NDCG@40, full embeddings

Fig. 7. Comparison of performance with/without influence embeddings.

weak-social-influence is made up of users with only 1 or 2 social-influence records, and the second group of strong-social-influence is made up of users with at least 10 social-influence records. We then visualize user embeddings of these two groups with t-SNE [43] (Fig. 9), which is a widely-used embedding-visualization tool [33]. From it, we can conclude that embeddings without CL at the right figure are more messy: two groups of embeddings are intertwined with each other. Therefore, CL assists to endow the strength of social influence into corresponding embeddings.

Combining the analysis of two aspects above, we indirectly demonstrate the effectiveness of CL for endowing semantics into embeddings.

4.4 Hyper-Parameters Study (RQ3)

In this section, we study the impact of some critical hyper-parameters on the Beibei dataset.

4.4.1 *Study of Weight Ratio Between $\mathcal{L}_{\mathbf{REC}}$ and $\mathcal{L}_{\mathbf{CL}}$.*

As mentioned in Section 3.5, we adopt the alternative training manner for model optimizing, thus the learning rates of

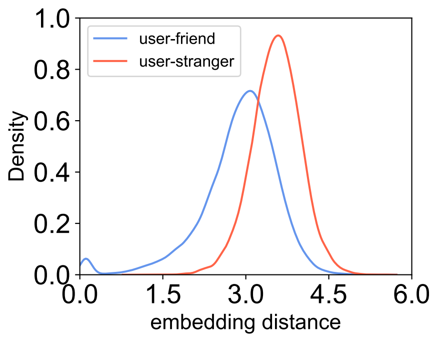


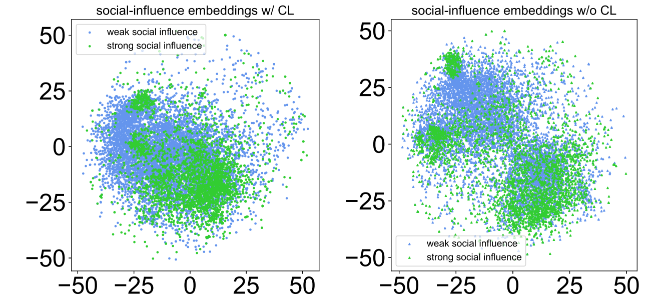
Fig. 8. Distance distribution of user-friend and user-stranger pairs.

$\mathcal{L}_{\mathbf{REC}}$ ($\zeta_{\mathbf{REC}}$) and $\mathcal{L}_{\mathbf{CL}}$ ($\zeta_{\mathbf{CL}}$) are important hyper-parameters. In our experiments, we have found a suitable choice of $\zeta_{\mathbf{REC}}$, 1e-4. Then to reduce the search cost, we fix it and adjust $\zeta_{\mathbf{CL}}$ to obtain the best performance. Since the learning rates is closely related to the importance of these two tasks, we define $w = \zeta_{\mathbf{CL}}/\zeta_{\mathbf{REC}}$ to indicate their weight ratio.[9]. We present the recommendation performance with different values of w in Fig. 10. Only results in $0 \sim 1$ are reported since the performance drops significantly when $w > 1$. As we can see, a suitable w is smaller than 1, demonstrating that the recommendation task is more important. In addition, the performance is steady when $0 \leq w \leq 0.3$ and achieves best when $w = 0.1$, thus the slower learning procedure of contrastive loss than that of recommendation loss will be a good choice. In other words, the large weight of contrastive learning will disturb normal model training and decrease overall performance. Moreover, the worse performance when $w = 0$ also demonstrates the necessity of introducing contrastive learning.

4.4.2 *Study of Embedding Size.*

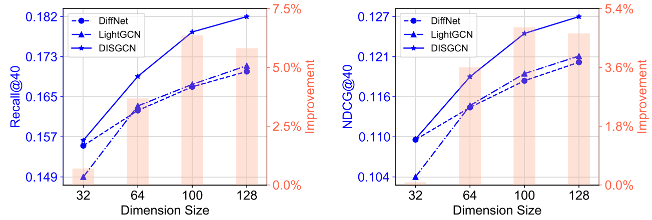
In representation learning, dimension size means model capacity [39]. In this section, we verify whether DISGCN always performs better than baselines with varying model capacity. Fig. 11 shows recommendation performances of CSR, DiffNet and our DISGCN with different embedding sizes. Note that DISGCN has a disentangled embedding layer with two embedding matrices, thus we use twice of embedding sizes for baselines for a fair comparison. For example, if 64-dimension is adopted for each embedding matrix of DISGCN, the embedding size of baselines will be

9. Please note that $w = 0$ means $\zeta_{\mathbf{CL}} = 0$, i.e., contrastive learning is excluded.



(a) embeddings with CL (b) embeddings without CL

Fig. 9. Visualization comparison between social-influence embeddings with and without contrastive learning.



(a) Recall (b) NDCG

Fig. 11. Study of embedding size on the Beibei dataset.

128. As we can see, the performance improvement of baselines becomes moderate gradually when embedding size increases. This can be explained as ability saturation of the model's preference learning. On the other hand, DISGCN obtains greater overall improvement compared with the other two baselines when embedding size becomes larger. More precisely, the relative improvement is [0.7%, 3.68%, 6.35%, 5.77%] and [0.08%, 3.60%, 4.88%, 4.69%] on Recall@40 and NDCG@40 respectively, when embedding size of baselines is [32, 64, 100, 128]. This demonstrates that modeling social influence becomes more important when there are enough model capacities for capturing user preferences. In other words, when preference modeling approaches the upper limit, our DISGCN can leverage another set of embeddings, influence embeddings, to model social influence and further improve recommendation accuracy.

5 Related Work

In this section, we introduce three categories of existing works related to this study and emphasize their limitations of leveraging social-relation data for modeling user behaviors.

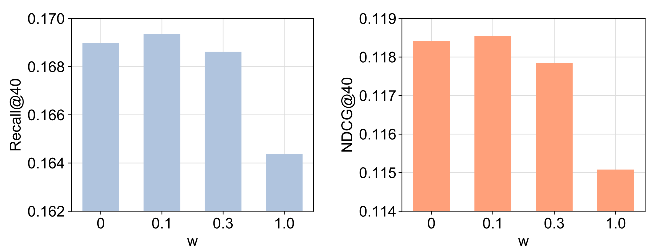
Social Recommendation. Social recommendation is generally defined as leveraging the social network for enhancing the recommendation. Most existing works of social recommendation [7], [9] are based on the social-homophily assumption that social-connected users are likely to have common interests. Specifically, there are two categories of methods implementing the assumption. The first category of works sets constraints on user representations to ensure enough similarity between that of social-connected users, via adopting regularization-based methods in the objective function [5], [7], [44] or smoothing-based information propagation [9], [11]. The second category of works [26], [45], [46] adopts multi-task learning to make the user representations shared across latent spaces of preference and social domains. Besides, some recent works [10], [12], [13], [39], [40] begun to study more complex social homophily in recommendation. Yu *et al.* [12] proposed that social homophily may do not apply for all the friends and developed generative adversarial network-based method to find these friends for robust and accurate preference learning. Yu *et al.* [10] also pointed out that social relations are noisy and heterogeneous in most cases of practice. Chen *et al.* [39] used attention networks to capture the various strength of homophily. Jin *et al.* [13] proposed a partial relationship aware information propagation scheme based on the assumption of homophily sparsity, with computationally efficient encoding design. Despite the effectiveness of these advances, we argue that modeling user interests based on social homophily is not enough to handle complex behaviors in real-world social scenarios.

Recently, Wu *et al.* [42] noticed the difference between social homophily and influence in social recommendation. However, the proposed method did not explicitly model these two effects and had no essential difference compared with the foregoing methods.

Graph Convolutional Networks for Recommendation. Recently, GCN-based models [21], [24], [47], [48] have become the state-of-the-art in recommender systems [49], built upon their strong power to extract high-order topology structure of graphs [50], [51]. The input data, such as user behavioral data, can be well represented via a graph structure, and then the recommendation task can be regarded as a link prediction task on the graph. PinSage [47] was the first attempt that applies GCN to item graphs with message passing layers, and it was deployed as the recommendation engine of Pinterest. Berg *et al.* [48] introduced the standard GCN to factorization user-item rating matrices for recommendation, which was further extended by [21] for recommender systems with implicit user behavioral data, achieving the state-of-the-art performance in most fundamental CF tasks. Besides these normal CF tasks, GCN-based methods have also performed well in many other recommendation tasks, including social recommendation [9], [11], knowledge graph-based recommendation [24], [52], attribute-aware recommendation [53], [54], sequential recommendation [55], diversified recommendation [56], etc.

In this work, we take advantage of GCN's strong representation ability and develop disentangled propagation mechanisms tailored to distinct traits of social homophily and influence for better representation learning.

Disentangled Representation Learning for Recommendation. In recommender systems, various factors take effect on user



(a) Recall (b) NDCG

Fig. 10. Study of the weight of contrastive learning on the Beibei dataset, where ζ_{REC} is model's learning rate.

behaviors. Thus, there are some recent works [57], [58], [59], [60] utilizing the disentangled representation learning for capturing independent factors. Bonner *et al*. [57] proposed to use two sets of embeddings to represent user preferences and exposure bias, which is further extended by [58] to disentangle user preferences and popularity bias. Wang *et al.* [59] proposed to introduces multiple input embeddings in graph convolutional networks to capture users' different intentions. Besides, disentanglement was adopted in sequential recommendation [61], in which seq2seq training strategy was proposed, and disentangled embeddings are utilized to model users' distinct intentions behind sequential behaviors. Despite the advances in various factors modeling, we argue that essential interpretability is not explored in corresponding model designs.

In this work, not only both social homophily-aware user interests and social influence are disentangled for social recommendation, but also explicit semantics of social influence are encoded in corresponding representations through contrastive learning.

6 Conclusion

In this work, from a novel and more reasonable perspective, we take both social homophily-aware user interests and social influence into consideration as two factors affecting user behaviors to approach the social recommendation task. Specifically, we propose a disentangled embedding layer that represents these two essential factors. Then tailored information propagation schemes are devised to disentangle personalized and item-specific neighborhood information of social homophily and social influence, respectively. We further propose the contrastive learning framework to extract explicit meanings from collected implicit social-influence behavioral data and inject them into disentangled influence embeddings. Extensive experiments and further studies verify the effectiveness of our proposed model and some key designs (i.e., social-influence embeddings and contrastive learning.). As for future works, we plan to conduct online A/B tests to evaluate the feasibility in real-world platforms further.

References

[1] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. World Wide Web Conf.*, 2017, pp. 173–182.

[2] S. Badrul, K. George, K. Joseph, and R. John, "Item-based collaborative filtering recommendation algorithms," in *Proc. Int. World Wide Web Conf.*, 2001, pp. 285–295.

[3] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, 2009, Art. no. 4.

[4] J. Schafer, D. Frankowski, J. Herlocke, and S. Sen, "Collaborative filtering recommender systems," in *The Adaptive Web*, Berlin, Germany: Springer, 2007, pp. 291–324.

[5] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 135–142.

[6] M. Jiang *et al.*, "Social contextual recommendation," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 45–54.

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

[8] 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.*, pp. 1–14, 2020.

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

[10] J. Yu, H. Yin, J. Li, M. Gao, Z. Huang, and L. Cui, "Enhancing social recommendation with adversarial graph convolutional networks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3727–3739, 2022.

[11] W. Fan *et al.*, "Graph neural networks for social recommendation," in *Proc. World Wide Web Conf.*, 2019, pp. 417–426.

[12] J. Yu, M. Gao, H. Yin, J. Li, C. Gao, and Q. Wang, "Generating reliable friends via adversarial training to improve social recommendation," in *Proc. IEEE Int. Conf. Data Mining*, 2019, pp. 768–777.

[13] B. Jin, K. Cheng, L. Zhang, Y. Fu, M. Yin, and L. Jiang, "Partial relationship aware influence diffusion via a multi-channel encoding scheme for social recommendation," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 585–594.

[14] M. Miller, S.-L. Lynn, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociol.*, pp. 415–444, 2001.

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

[16] L. Ma, R. Krishnan, and A. L. Montgomery, "Latent homophily or social influence? An empirical analysis of purchase within a social network," *Manage. Sci.*, vol. 61, no. 2, pp. 454–473, 2015.

[17] K. Kee-Young and X. Ge, "The effects of social media on e-commerce: A perspective of social impact theory," in *Proc. Hawaii Int. Conf. Syst. Sci.*, 2012, pp. 1814–1823.

[18] A. Anagnostopoulos, R. Kumar, and M. Mahdian, "Influence and correlation in social networks," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2008, pp. 7–15.

[19] K. Lewis, M. Gonzalez, and J. Kaufman, "Social selection and peer influence in an online social network," *Proc. Nat. Acad. Sci., USA*, vol. 109, no. 1, pp. 68–72, 2012.

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

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

[22] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1725–1735.

[23] K. Huang, C. Xiao, L. M. Glass, M. Zitnik, and J. Sun, "SkipGNN: Predicting molecular interactions with skip-graph networks," *Sci. Rep.*, vol. 10, no. 1, pp. 1–16, 2020.

[24] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proc. SIGKDD Conf. Knowl. Discov. Data Mining*, 2019, pp. 950–958.

[25] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 452–461.

[26] C. Chen, M. Zhang, C. Wang, and W. Ma, "An efficient adaptive transfer neural network for social-aware recommendation," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 225–234.

[27] E. Palumbo, G. Rizzo, R. Troncy, E. Baralis, M. Osella, and E. Ferro, "Knowledge graph embeddings with Node2Vec for item recommendation," in *Proc. Eur. Semantic Web Conf.*, 2018, pp. 117–120.

[28] C. Cheng, H. Yang, I. King, and M. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in *Proc. AAAI Conf. Artif. Intell.*, 2012, pp. 17–23.

[29] W. Krichene and S. Rendle, "On sampled metrics for item recommendation," in *Proc. SIGKDD Conf. Knowl. Discov. Data Mining*, 2020, pp. 1748–1757.

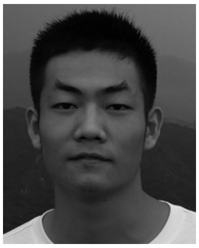
[30] C. Gao *et al.*, "Social recommendation with characterized regularization," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 6, pp. 2921–2933, 2022.

[31] W. Fan *et al.*, "Graph neural networks for social recommendation," in *Proc. World Wide Web Conf.*, 2019, pp. 417–426.

[32] J. Ma, P. Cui, K. Kuang, X. Wang, and W. Zhu, "Disentangled graph convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4212–4221.

[33] Y. Zheng, C. Gao, X. Li, X. He, Y. Li, and D. Jin, "Disentangling user interest and conformity for recommendation with causal embedding," in *Proc. Web Conf.*, 2021, pp. 2980–2991.

[34] H. Wang, H. Shen, W. Ouyang, and X. Cheng, "Exploiting poi-specific geographical influence for point-of-interest recommendation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 3877–3883.
[35] Z. Cheng, Y. Ding, X. He, L. Zhu, X. Song, and M. S. Kankanhalli, "A^3 NCF: An adaptive aspect attention model for rating prediction," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 3748–3754.
[36] H.-J. M. Shi, D. Mudigere, M. Naumov, and J. Yang, "Compositional embeddings using complementary partitions for memory-efficient recommendation systems," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 165–175.
[37] M. R. Joglekar *et al.*, "Neural input search for large scale recommendation models," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 2387–2397.
[38] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
[39] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Social attentional memory network: Modeling aspect-and friend-level differences in recommendation," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, 2019, pp. 177–185.
[40] 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.
[41] W. Fan, Q. Li, and M. Cheng, "Deep modeling of social relations for recommendation," in *Proc. AAAI Conf. Artif. Intell.*, 2018, Art. no. 1029.
[42] Q. Wu *et al.*, "Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems," in *Proc. World Wide Web Conf.*, 2019, pp. 2091–2102.
[43] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.
[44] X. Wang, X. He, L. Nie, and T.-S. Chua, "Item silk road: Recommending items from information domains to social users," in *Proc. Int. Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 185–194.
[45] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2008, pp. 931–940.
[46] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, "Bridging collaborative filtering and semi-supervised learning: A neural approach for POI recommendation," in *Proc. SIGKDD Conf. Knowl. Discov. Data Mining*, 2017, pp. 1245–1254.
[47] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for Web-scale recommender systems," in *Proc. SIGKDD Conf. Knowl. Discov. Data Mining*, 2018, pp. 974–983.
[48] R. v. d. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," 2017, *arXiv:1706.02263*.
[49] C. Gao *et al.*, "Graph neural networks for recommender systems: Challenges, methods, and directions," 2021, *arXiv:2109.12843*.
[50] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–14.
[51] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–12.
[52] X. Wang *et al.*, "Learning intents behind interactions with knowledge graph for recommendation," in *Proc. Web Conf.*, 2021, pp. 878–887.
[53] B. Liu, R. Tang, Y. Chen, J. Yu, H. Guo, and Y. Zhang, "Feature generation by convolutional neural network for click-through rate prediction," in *Proc. World Wide Web Conf.*, 2019, pp. 1119–1129.
[54] Y. Zheng, C. Gao, X. He, Y. Li, and D. Jin, "Price-aware recommendation with graph convolutional networks," in *Proc. IEEE 36th Int. Conf. Data Eng.*, 2020, pp. 133–144.
[55] J. Chang *et al.*, "Sequential recommendation with graph convolutional networks," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 378–387.
[56] Y. Zheng, C. Gao, L. Chen, D. Jin, and Y. Li, "DGCN: Diversified recommendation with graph convolutional networks," in *Proc. Web Conf.*, 2021, pp. 401–412.
[57] S. Bonner and F. Vasile, "Causal embeddings for recommendation," in *Proc. 12th ACM Conf. Recommender Syst.*, 2018, pp. 104–112.
[58] Y. Zheng, C. Gao, X. Li, X. He, Y. Li, and D. Jin, "Disentangling user interest and popularity bias for recommendation with causal embedding," in *Proc. World Wide Web Conf.*, 2021, pp. 2980–2991.
[59] X. Wang, H. Jin, A. Zhang, X. He, T. Xu, and T.-S. Chua, "Disentangled graph collaborative filtering," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 1001–1010.
[60] Y. Zheng *et al.*, "Disentangling long and short-term interests for recommendation," in *Proc. ACM Web Conf.*, 2022, pp. 2256–2267.
[61] J. Ma, C. Zhou, H. Yang, P. Cui, X. Wang, and W. Zhu, "Disentangled self-supervision in sequential recommenders," in *Proc. SIGKDD Conf. Knowl. Discov. Data Mining*, 2020, pp. 483–491.



Nian Li received the bachelor's degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2021. He is currently working toward the PhD degree with the Graduate School at Shenzhen, Department of Electronic Engineering, Tsinghua University, Shenzhen, China. His research interests include on recommender systems and and graph learning.

Chen Gao received the bachelor's and PhD degrees from the Department of Electronic Engineering, Tsinghua University, in 2016 and 2021, respectively. He is now a postdoctoral researcher with the Department of Electronic Engineering, Tsinghua University. His research interests include data mining and information retrieval, especially on recommender system. He has more than 30 publications in journals and conferences such as *IEEE Transactions on Knowledge and Data Engineering*, *ACM Transactions on Knowledge Discovery from Data*, SIGIR, WWW, ICLR, KDD, IJCAI, etc. His work on GNN-based bundle recommendation received the Best Short Paper Honorable Mention Award in SIGIR 2020. He is also selected as Top 100 Chinese New Stars in Artificial Intelligence (Data Mining Area) by Baidu Scholar.

Depeng Jin (Member, IEEE) received the BS and PhD degrees in electronics engineering from Tsinghua University, Beijing, China, in 1995 and 1999, respectively. He is currently an professor with Tsinghua University and vice chair of Department of Electronic Engineering. He was awarded National Scientific and Technological Innovation Prize (Second Class), in 2002.

Qingmin Liao (Senior Member, IEEE) received the BS degree in radio technology from the University of Electronic Science and Technology of China, Chengdu, China, in 1984, and the MS and PhD degrees in signal processing and telecommunications from the University of Rennes 1, Rennes, France, in 1990 and 1994, respectively. Since 1995, he has been joining with Tsinghua University, Beijing, China. In 2002, he became professor with the Department of Electronic Engineering, Tsinghua University. Since 2006, he has been the director of the Shenzhen Key Laboratory of Information Science and Technology, in the Graduate School at Shenzhen, Tsinghua University. Over the last 35 years, he has published more than 200 peer-reviewed journal and conference papers. His research interests include image/video processing, transmission and analysis; biometrics; and their applications to teledetection, medicine, industry, and sports.