

Social capital matters: Towards comprehensive user preference for product recommendation with deep learning

Weiye Li^{a,1}, Ming Gao^{a,*,1}, Bowei Chen^{b,1}, Jingmin An^{a,1}, Yeming Gong^{c,1}

^a School of Management Science and Engineering, Key Laboratory of Big Data Management Optimization and Decision of Liaoning Province, Dongbei University of Finance and Economics, China

^b Adam Smith Business School, University of Glasgow, United Kingdom

^c Artificial Intelligence in Management Institute, EMLYON Business School, France

ARTICLE INFO

Keywords:

Social recommendation
Social capital
Design science
Deep learning
Online decision-making

ABSTRACT

Social recommender systems help address data sparsity in user-product interactions by leveraging social relationships to infer user preferences. However, existing models often overlook the role of social capital that influence decision-making in social commerce. Social capital consists of structural, relational, and cognitive dimensions, all of which shape user preferences. To better understand these influences, we propose a multi-task learning framework named DeepSC that integrates social capital theory into preference modeling. Its user preference learning module extracts structural features through graph-based pre-training, learns relational features from dynamic user embeddings, and models cognitive features using a hypergraph attention network. Additionally, the dual graph-based product feature learning module enhances cognitive feature extraction by incorporating product co-interactions. DeepSC is optimized through a joint learning objective, combining point-wise and pair-wise learning with an auxiliary social link prediction task to refine user representations. Experiments on three e-commerce datasets demonstrate that DeepSC significantly outperforms the state-of-the-art recommendation models, highlighting the effectiveness of integrating social capital into social preference learning. Our research advances social recommendation by providing a social capital theory-driven approach to modeling user behavior in digital commerce.

1. Introduction

Over the past decade, recommender systems have played an essential role in social commerce platforms by leveraging user-product interactions and social relationships to alleviate data sparsity and enhance consumer decision-making [1]. TikTok, one of the most popular social commerce platforms, has surpassed 2.6 billion downloads worldwide and 100 million users in the US [2], with a valuation of over \$100 billion as reported by Wedbush analyst Dan Ives.² The phenomenal growth of TikTok is largely driven by its recommender system, valued at over \$50 billion, which tailors content to each user's specific interests and influences their media consumption behaviors.² As another successful social platforms, Pinterest's recommender systems account for more than 80% of total user engagement [3] and result in a 25% increase in impressions for the platform's 'Shop the Look' product.³

Social recommender systems have also attracted considerable interest within the academic community [4]. As a foundational work, Arazy

et al. [5] explore the factors that affect the willingness to accept recommendations, such as homophily, tie strength, and trustworthiness. Further research [6] shows that latent homophily and social influence promote the similar purchase decisions among the users and their social friends. Building on these empirical research findings, considerable efforts have been directed toward capturing homophily and social influence to derive social preferences using matrix factorization [7] and graph learning [8]. Particularly, recent social recommenders focus on enhancing the quality of user representations by filtering out noise from excessive redundant social connections [9,10]. Despite some advantages, most works fail to extract multiple user preferences before social preference learning, leading to a biased intention inference, as shown in Fig. 1. Intuitively, users have multiple motivations to ask for some advices from different friends. For example, a girl may seek advice on dressing from her best friend, who happens to be a fashionista. We suggest that capturing multiple user preferences has the potential to enhance the social preference learning.

* Correspondence to: No. 217 JianShan St., Shahekou District, Dalian, PR China.

E-mail address: gm@dufe.edu.cn (M. Gao).

¹ These authors contributed equally to this work.

² <https://www.cbsnews.com/news/how-much-is-tiktok-worth-2024/>.

³ <https://www.socialmediatoday.com/news/pinterests-recommendation-algorithms-are-improving-which-is-important-to/543376/>.

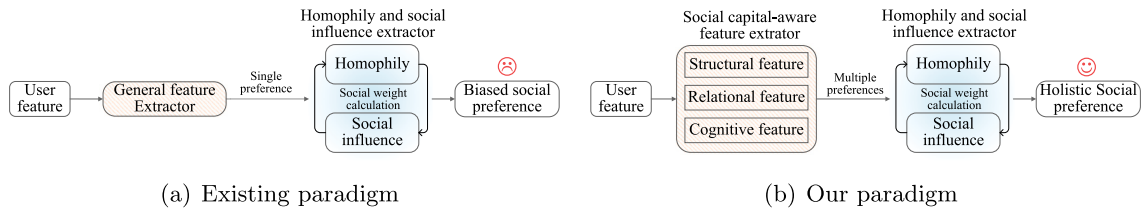


Fig. 1. The comparison of social preference learning process of various solutions.

In response to the identified research gap, this study aims to answer the following two key research questions. First, which aspects of user preferences are most valuable for social preference learning? Existing approaches employ disentangled learning [11,12] and clustering methods [13] to extract various user preferences. However, these works focus on technical modeling and analyzing, overlook the crucial role of theory in model design. While performance-driven models are valuable, aligning recommendation models with user intentions enhances both understanding and acceptance [14]. Second, how to capture these different user preferences and integrate them into a unified user representation? Since various aspects of user preferences exhibit unique characteristics, applying a uniform approach across all aspects can lead to suboptimal performance [11,12]. Thus, it is essential to design tailored methodologies for different facets of user preferences.

To tackle these research questions, we propose a deep learning-based social recommender system named DeepSC, which incorporates the impact of social capital in online decision-making. Social capital theory suggests that social network provides actual or potential resources to users [15], highlighting how interpersonal influences shape preferences [16]. This theory provides a promising guideline to model user features from three specific dimensions [17]: structural (user's position within the social network), relational (nature and quality of relationships), and cognitive (shared understandings) [15]. It is suggested that these multiple preferences have their potential to improve the social preference learning by integrating information about user's relationships. Accordingly, we design a social capital-driven feature extractor to learn user representations grounded in these dimensions. Specifically, the structural dimension focuses on positional information within an ego network, the relational dimension emphasizes stable interpersonal characteristics represented as continuously updated user embedding, and the cognitive dimension targets higher-order features [18] associated with product consensus. The three-dimensional user features reveal the preference learning process behind recommendations, improving the belief and acceptance to recommender systems. In the subsequent social preference learning process, the gated attention network [19] is adopted to capture the influence of each social friend, taking advantage of the MLP-based and dot-product attention mechanisms. Furthermore, an auxiliary social link prediction task is introduced to refine influence assignments from the gated attention network, enhancing the representation capacity of user features.

Beyond user feature modeling, product feature learning plays a critical role in user-product matching. Since cognitive dimension features depend on user-product interactions, we propose a dual graph-based approach to learn product representations. Specifically, we construct a product-oriented interaction hypergraph, where hyper-nodes represent products and hyper-edges denote groups of products engaged by the same user. A hyper-attention network [20] is then adopted to learn high-order product features. Inspired by [21], we also introduce a co-interacted product network, derived from user-product interactions, to illustrate co-occurrence relationships between products engaged by the same users. Subsequently, we adopt the light graph convolution network (LightGCN) [22] to efficiently capture the complex latent dependencies between products. Notably, the product feature learning module of DeepSC does not rely on external side information, such as product categories or brands, enhancing its efficiency. Moreover,

we combine point-wise [23,24] and pair-wise [18,25] learning objectives to train our model, which simultaneously enhances calibration and ranking capabilities. This joint learning task ensures that the recommendations align with actual user click-through behavior while optimizing product rankings [26,27].

This research makes four key contributions to decision support systems and social recommendation. First, we highlight the essential role of social capital in guiding user feature learning for better social preference modeling. Second, we propose a method for parameterizing the three dimensions of social capital to align this theory with model design, potentially providing a solid foundation for future studies on the impact of social capital in different contexts. Third, we develop a dual graph-based product feature learning module that leverages hypergraph structures and co-interacted product networks to enhance the feature learning of the cognitive dimension. Finally, we introduce a novel multi-task learning objective to enhance the representativeness of social capital-aware features, which consists of an auxiliary social link prediction task and a main joint learning task. The social link prediction task is designed to refine social preference modeling, while the joint learning task balances calibration and ranking to provide robust and effective decision support. Extensive experiments on three real-world datasets, i.e., Ciao, Epinions, and Yelp, validate the superiority of DeepSC over 19 baseline models, with ablation studies confirming the importance of its core components.

The rest of the paper is structured as follows. Section 2 reviews related literature. Section 3 defines the social recommendation problem, outlines the DeepSC architecture, and details its technical components. Section 4 describes the experimental setup, results, and analysis. Finally, Section 5 discusses research contributions, practical implications, and future directions.

2. Related work

This section demonstrates the effectiveness of representation learning and examines key studies on social recommender systems, highlighting research trends and our study's motivation. Finally, we introduce relevant theoretical frameworks, particularly social capital theory, and explain their role in guiding our model design.

2.1. Social recommender systems based on representation learning

Research in recommender systems has made impressive advances in supporting users' online decision-making by alleviating information overload [14,18]. Early efforts rely on feature engineering to capture the features of users and products from rich external side information, such as user profiles and product attributes [28]. However, these approaches face an effectiveness-efficiency trade-off: while simplistic feature engineering often yields poor performance, more sophisticated techniques tend to introduce substantial computational overhead. Additionally, obtaining rich side information can be challenging in practice—particularly for privacy-conscious users or dynamic products with frequently changing attributes [29].

To address these challenges, the representation learning [30] has been widely adopted by recommender systems for end-to-end feature extraction. This approach represents a user or a product with a latent vector to capture its characteristics. As an effective tool widely utilized

in deep learning, the key idea behind representation learning is to seek a low-dimensional embedding of the data while preserving various discriminative factors of variation inherent in the data. Therefore, we can effectively and efficiently learn the features of users and products without relying on any explicit attributes, using only their IDs.

As a promising subfield of recommender systems, social recommendation leverages the social relationships to improve recommendation performance, which assumes that the opinions from the social friends impact the focal user's decision-making. Technically, most early works [31,32] capture user preferences from social domain by adopting joint matrix factorization (MF) and adding social-aware regularization terms to the objective function. Specifically, SocialMF [31] incorporated social influences from friends into the MF-based model for rating prediction. Moreover, SoReg [32] considered social factors as regularization terms to constrain the MF, which is conducive to improving the precision of recommendations. These methods often fail to distinguish the varying influence of individual friends and are limited in their ability to capture signals from higher-order social connections, which leads to suboptimal performance. Additionally, few efforts [8,33] transform user-product interactions and social relationships into user-product bipartite graph and social network and adopt random walk-based approaches to capture user preference. For example, CUNE [33] learned implicit interests and identifies semantic friends by biased random walk and skip-gram, which is used to complement sparse explicit social relations. Yet, such methods tend to capture unreliable or noisy implicit social connections, often resulting in biased user preference learning.

Recently, graph neural networks (GNNs) have been widely adopted to explicitly model the users' latent preferences with information diffusion process in the social network. Specifically, GCN-based methods, such as DiffNet [34] and SocialLGN [35], efficiently capture user preferences from interaction domain and social domain. Moreover, GAT-based methods, such as GraphRec [36] and DANSER [37], adopt soft denoising approaches based on attention networks to mitigate the impact of noisy data by adaptively assigning lower weights to uninterested products and unreliable social friends. Furthermore, GDMRS [9] and MADM [10] adopt hard denoising approaches to filter out noisy interactions and social relationships. However, these hard denoising approaches lead to sparser data, which hinders the ability to effectively learn user preferences. To address data sparsity, GSFR [38] and DICER [23] generate pseudo social relationships based on similar interactions, then capturing opinions from high-order social friends. Despite these advances, many of these approaches emphasize homogeneous preferences and overlook the role of heterogeneous opinions in user decision-making. In addition, the prevalent use of stacking-based GNN architectures often leads to over-smoothing, where node features converge to indistinguishable values across layers, thereby degrading model expressiveness [39].

A further challenge arises from the representation of user-product interactions. Most social recommender systems model these interactions using a heterogeneous bipartite graph. However, such representations are inherently limited to pairwise relations, which restricts their capacity to capture group-level relationships directly [40]. Hypergraph learning has emerged as a promising paradigm to address this limitation. By leveraging flexible hyperedges, it facilitates information propagation beyond pairwise interactions. For instance, in an item-oriented hypergraph, a hyperedge can connect all users who interact with a particular item, thereby defining the item's precise audience group. Recent studies, such as [18,41], have demonstrated the superiority of hypergraph-based approaches over bipartite graphs in learning user and product representations for recommendation. Moreover, we employ a hypergraph attention network to replace conventional hypergraph convolutional networks. This model learns a dynamic incidence matrix that adaptively captures nuanced relationships between users and products based on feature affinities.

2.2. Empirical and theoretical underpinnings of DeepSC

The traditional theoretical foundations of social recommender systems are social influence theory and homophily theory. The two theories are introduced to support the calculation and propagation of user-pair similarity. Specifically, social influence theory suggests that users in social networks are influenced by the attitudes or behaviors of their social friends, leading them to make decisions similar to those friends. Some studies [42,43] showed that social influence is often more effective than the similarity of historical interactions in inferring users' intentions. Moreover, homophily theory suggests that users are more likely to form social relationships with those who share similar characteristics, such as demographics or common interests [44]. However, existing social recommender systems disregard the influence of heterogeneous friends who provide diverse opinions. Furthermore, it remains unclear which factors specifically influence the calculation of user-pair similarity.

Despite existing works acknowledging the importance of social influence and homophily in social recommendations, they often overlook the influence of social capital, which has demonstrated its relevance in knowledge sharing and decision-making [45]. Therefore, we argue that social capital plays a significant role in social recommendations and introduce it to guide the design of our DeepSC. Social capital refers to the value of all the resources and benefits that an individual can obtain and control through their social relationships [46]. It not only establishes close relationships between users, but also encourages communication, identification, and trust [47]. In this paper, we follow the widely adopted measurement of social capital from three particular dimensions [17]. Specifically, structural dimension concerns the position in the social network, while the relational dimension refers to those assets created and leveraged through long-term relationships. Moreover, the cognitive dimension comprises shared values developed when users have common opinions on various products. To sum up, the social capital theory offers a valuable theoretical foundation for capturing fine-grained user preferences [17]. In this paper, with the social capital-aware diffusion of social influence and homophily, our DeepSC extends existing methods to capture more precise opinions from social friends.

Moreover, we extend the concept of user homophily to introduce the concept of product homophily, assuming that products interacted by the same users have similar attractiveness when exposed to other users. In this paper, we extract the co-interacted product network from user-product interactions, where the edges represents the co-interacted relations and the weights represents the normalized co-interacted frequencies. Product homophily analysis is expected to provide appropriate recommendations by finding what most customers prefer to [21]. It can also enhance the feature learning for niche products by analyzing co-interacted popular products.

3. DeepSC

Traditionally, rating prediction has been the primary task in social recommendation. However, recent research [48] suggest that top-K recommendation provides greater business value by enhancing customer engagement and advertising revenue. Formally, let $U = \{u_1, \dots, u_{|U|}\}$ be the set of users and $V = \{v_1, \dots, v_{|V|}\}$ be the set of products. The interaction matrix $R \in \mathbb{R}^{|U| \times |V|}$ records user-product interactions, where $r_{ij} = 1$ indicates that user i interacts with product j , otherwise $r_{ij} = 0$. Similarly, the social link matrix $S \in \mathbb{R}^{|U| \times |U|}$ represents social relationships, where $s_{ij} = 1$ if user u_i follows user u_j , and $s_{ij} = 0$ otherwise. Given R and S , our goal is to predict the interaction probability $\hat{r}_{ij}$ between a user u_i and a candidate product v_j , ranking products based on their likelihood of interaction.

As illustrated in Fig. 2, our proposed framework, DeepSC, consists of three key components. The user feature learning module, inspired by

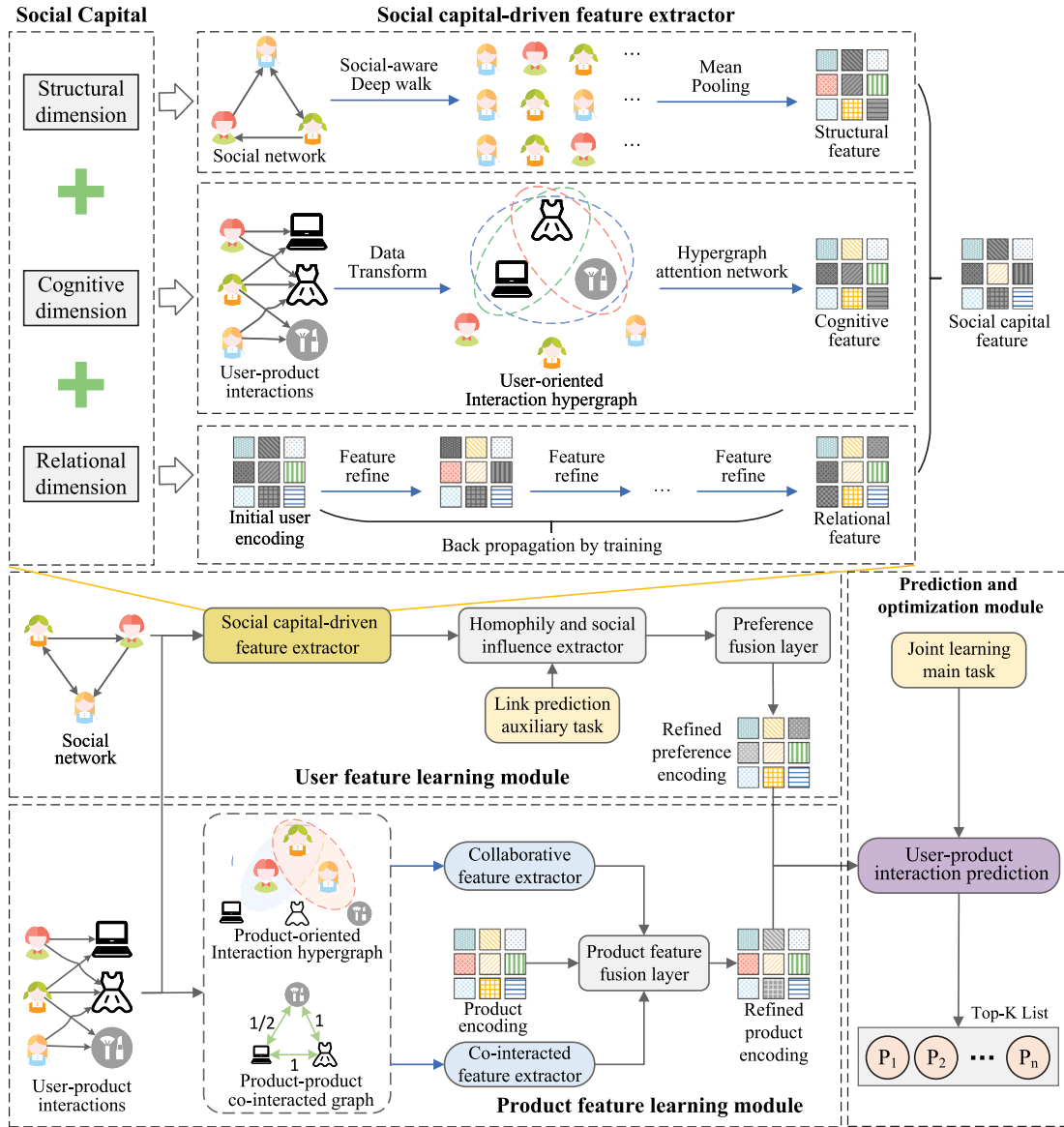


Fig. 2. Graphical overview of our proposed modeling framework.

social capital theory, extracts structural, relational, and cognitive features from user interactions and social relationships. The product feature learning module captures product engagement patterns using user-product interactions and product-product co-interaction networks, enhancing feature representation. The prediction and optimization module refines user-product matching through a multi-task learning objective, incorporating both ranking-based optimization and an auxiliary learning task. By integrating social capital dimensions, DeepSC effectively enhances user preference modeling, leading to improved recommendation performance in social commerce environments.

3.1. User feature learning module

The user feature learning module aims to capture the comprehensive user representations. Specifically, the social capital-driven feature extractor learns the three dimensional user features according to the social capital theory. Subsequently, the homophily and social influence extractor updates the user features by capturing the opinions from their social friends with the diffusion of social influence. After that, the user feature fusion layer integrates various user features derived from the two user feature extractors to obtain comprehensive user representations.

3.1.1. Social capital-driven feature extractor

According to the social capital theory, it is rational to model user preferences from structural dimension, relational dimension, and cognitive dimension. Specifically, considering that social network structures significantly impact the recommendations provided by agents [16], we propose the random walk-based structural pre-training on ego-networks to parameterize the structural dimensional features. Formally, on the focal user's ego-network $S_u \subset S$, the transition probability from the focal user node n_u to her social friend n_j during a random walk is given by $P_{n_i \rightarrow n_u}(n_j | n_u) = \alpha_{pq}(n_i, n_j)$, where $\alpha_{pq}(n_i, n_j)$ is a normalization factor that depends on the previous user node n_i and the next candidate node n_j . Moreover, a larger p encourages to visit the user nodes that are semantically related but structurally distant, while a larger q encourages to visit the user nodes that are directly connected to the focal user node. After generating the user node sequences, the structural features $St_u \in \mathbb{R}^d$ of n_u can be learned by Skip-gram [49].

Inspired by DGRc [50], the free user embedding $e_u^{init} \in \mathbb{R}^d$ is adopted to represent the relational dimensional features Re_u of the focal user u , focusing on long-term user characteristics, where d denotes the embedding size.

Moreover, we capture the consensus from user-product interactions to parameterize the features of the cognitive social capital. Specifically,

we introduce a hypergraph-based method to extract the complex hidden relations among users and their interacted products [18]. Formally, the user-oriented interaction hypergraph is denoted as $HG_u = (\mathcal{V}_u, \mathcal{E}_v)$ with $|\mathcal{U}|$ hyper-nodes and $|\mathcal{V}|$ hyper-edges, where the hyper-node $hn_i \in \mathcal{V}_u$ represents the user u_i and the hyper-edge $he_j \in \mathcal{E}_v$ represents the product v_j . Different from traditional user-product bipartite graph merely focuses on pair-wise interactive relationships, the hyper-edge of our user-oriented interaction hypergraph simultaneously connects multiple hyper-nodes to learn the complex hidden interactive features. After that, the multi-head hypergraph attention network [20] is adopted to capture users' multi-aspect dynamic consensus on the interacted products. Formally, the one-aspect features of the cognitive social capital are learned as follows:

$$X_u^{(l+1)} = D_u^{-1} H_u W_u B_v^{-1} H_v^T X_u^{(l)} P_u, \quad (1)$$

$$H_u^{ij} = \text{Softmax} \left(\phi \left(q_u^T \left[x_i^u P_u \parallel x_j^v P_u \right] \right) \right), \quad (2)$$

where $D_u \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{U}|}$ and $B_v \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ denote the diagonal degree matrices of the user hyper-nodes and product hyper-edges, indicating the user engagement and the product popularity, respectively. The incidence matrix $H_u \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{V}|}$ represents the user-product interactive relationships. The weight matrix $P_u \in \mathbb{R}^{d \times d}$ captures the degree-normalized weights of user-product interactions, mitigating the excessive influence of highly active users. $x_i^u \in \mathbb{R}^d$ and $x_j^v \in \mathbb{R}^d$ denote the features of user hyper-node and product hyper-edge, respectively. We follow the design in [51], employing Gaussian Error Linear Units (GeLU) as the non-linear activation function, denoted by $\phi(\cdot)$. $W_u \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ and $q_u \in \mathbb{R}^{2d}$ are learnable parameters. $[\cdot \parallel \cdot]$ denotes the concatenating operation. For briefly, the above mentioned process is denoted as $X_u^{(l+1)} = HAN^{(l)}(X_u^{(l)}, \theta_u^l)$, where θ_u^l is the parameter set of l th layer. Furthermore, the multi-head mechanism is adopted to capture the diverse intentions of users when interacting with products. Just as we consider brand, style, and price when buying clothes. Formally,

$$X_u^{(l+1)} = MLP_{mh}^u \left(\left[HAN_1^{(l)}(X_u^{(l)}, \theta_u^l) \parallel \dots \parallel HAN_f^{(l)}(X_u^{(l)}, \theta_u^l) \right] \right), \quad (3)$$

where f denotes the number of attention heads. $MLP_{mh}^u(\cdot)$ denotes the 3-layer MLP of the form $\mathbb{R}^d \rightarrow \mathbb{R}^d$. Finally, we summary the learned features from each layer to capture the features of the cognitive social capital $C_{o_u} \in \mathbb{R}^d$: $C_{o_u} = \frac{1}{L} \sum_{l=1}^L X_u^{(l)}$, where L denotes the number of stacking layers. After learning the three dimensional features, it is necessary to fuse them for capturing the comprehensive social capital features. Here we adopt the simple sum operation to improve the training efficiency: $S_{c_u} = St_u + Re_u + C_{o_u}$.

It is noted that while most previous studies [52–54] relied on questionnaires to measure social capital dimensions, they faced the challenges of self-report bias, low response rates, and temporal constraints. By contrast, our end-to-end parameterized approach automatically extracts complex patterns from large datasets and adaptively optimizes learned features, providing a dynamic and efficient alternative for feature scaling and modeling.

3.1.2. Homophily and social influence extractor

After capturing the social capital features, the homophily and social influence extractor aims to gather the opinions from users' social friends for learning the social-aware user preferences. Specifically, the social network is defined as $G_s = (\mathcal{V}_u, \mathcal{E}_s)$, where the $\mathcal{V}_u$ is the user node set and the $\mathcal{E}_s$ is the link set of the social relationships. Subsequently, the gated attention network is introduced to learn the representations of the social preferences by gathering the opinions from the social friends. Formally,

$$\hat{h}_i^u = W_1 \alpha_{i,i} h_i^u + \sum_{j \in N_S(i)} W_2 \alpha_{i,j} h_j^u, \quad (4)$$

$$\alpha_{i,j} = \text{Softmax} \left(\omega \left(q_s^T \left[W_3 h_i^u \parallel W_4 h_j^u \right] \cdot \sigma \left((W_3 h_i^u)^T W_4 h_j^u \right) \right) \right), \quad (5)$$

where h_i^u denotes the features of the user u_i initialized to S_{c_u} . $N_S(i)$ denotes the neighborhood social friend set of the user u_i . $\omega(\cdot)$ denotes

the non-linear activation function named LeakyReLU with 0.2 slope. $\sigma(\cdot)$ denotes the sigmoid activation function. $W_1, W_2, W_3, W_4 \in \mathbb{R}^d$, and $q_s \in \mathbb{R}^{2d}$ denote the learnable parameters. The gated attention network benefits from both the MLP-based attention mechanism and the dot-product attention mechanism [55], improving the capacity to gather unbiased opinions from different social friends.

To further enhance the denoising capacity, we introduce an auxiliary social link prediction task aimed to calibrate the learned attention weight for each social friend. Formally,

$$\mathcal{L}_{aux}^{(l)} = -\frac{1}{\mathcal{E}_s \cup \mathcal{E}_s^-} \times \sum_{(i,j) \in \mathcal{E}_s \cup \mathcal{E}_s^-} \left[\mathbb{I}(y_{ij} = 1) \cdot \log(\hat{y}_{ij}^{(l)}) + \mathbb{I}(y_{ij} = 0) \cdot \log(1 - \hat{y}_{ij}^{(l)}) \right], \quad (6)$$

$$\hat{y}_{ij} = \sigma \left((W_3 h_i^u)^T W_4 h_j^u \right), \quad (7)$$

where $\mathcal{E}_s^-$ is the negative social link set sampling from the complementary set of $\mathcal{E}_s$. $\mathbb{I}(\cdot)$ denotes the indicator function, where $\mathbb{I} = 1$ if there is a social link between u_i and u_j , otherwise $\mathbb{I} = 0$.

Similar to Eq. (3), the multi-head attention mechanism is adopted to capture the multi-aspect opinions from her social friend. For example, we may choose to follow our fashion-savvy friends' preferences for brands and styles when purchasing clothing. After capturing the diverse social opinions, we stack the gated attention layer to stimulate the propagation process of social influence. Subsequently, the social-aware user feature is captured by aggregating the user features at each layer as follows: $S_{p_i} = \frac{1}{L} \sum_{l=1}^L h_i^{u,(l)}$. Similarly, the total loss of the auxiliary task is computed by summing the loss at each layer as follows: $\mathcal{L}_{aux} = \frac{1}{L} \sum_{l=1}^L \mathcal{L}_{aux}^{(l)}$.

3.1.3. User feature fusion layer

To learn better user representations, the comprehensive user representations are learned by fusing the social capital features and the social-aware user features. Considering that the pre-trained features are frozen for efficiency, we disregard the structural dimensional features of social capital when learning the dynamic user representations. Formally, $e_i^u = MLP_u[Re_i \| C_{o_i} \| S_{p_i}]$, where $MLP_u(\cdot)$ denotes the 3-layer MLP of the form $\mathbb{R}^{3d} \rightarrow \mathbb{R}^d$.

3.2. Product feature learning module

The product feature learning module aims to capture the comprehensive product representations. Specifically, the collaborative feature extractor captures the collaborative signals to learn the product features from the product-oriented interaction hypergraph. Subsequently, the co-interacted feature extractor captures the co-occurrence signals to learn the product features from the co-interacted product network. After that, the product feature fusion layer integrates various product representations derived from the two product feature extractors to obtain comprehensive product representations.

3.2.1. Collaborative feature extractor

The collaborative feature extractor first constructs the product-oriented interaction hypergraph based on the product-user interactions to represent the group-level interacted relations. Formally, the product-oriented interaction hypergraph is denoted as $HG_v = (\mathcal{V}_v, \mathcal{E}_u)$ with $|\mathcal{V}|$ hyper-nodes and $|\mathcal{U}|$ hyper-edges, where the hyper-node $hn_i \in \mathcal{V}_v$ represents the product v_i and the hyper-edge $he_j \in \mathcal{E}_u$ represents the user u_j . Subsequently, the multi-head hypergraph attention network is adopted to capture product's multi-aspect characteristics by gathering the collaborative opinions from the users interacted with the focal product. Similar to Eq. (3),

$$X_v^{(l+1)} = MLP_{mh}^v \left(\left[HAN_1^{(l)}(X_v^{(l)}, \theta_v^l) \parallel \dots \parallel HAN_f^{(l)}(X_v^{(l)}, \theta_v^l) \right] \right), \quad (8)$$

where θ_v^l is the parameter set of l th layer. $MLP_{mh}^v(\cdot)$ denotes the 3-layer MLP of the form $\mathbb{R}^d \rightarrow \mathbb{R}^d$. Furthermore, we integrate the product features from different layers to learn the final collaborative product features as follows: $c_{f_v} = \frac{1}{L} \sum_{l=1}^L X_v^{(l)}$.

3.2.2. Co-interacted feature extractor

Inspired by market basket analysis, the co-interacted feature extractor focuses on discovering user behavioral patterns by capturing the co-occurrence relationships among products, which improves the quality of the product features. Formally, the co-interacted product network is denoted as $G_v = (\mathcal{V}_v, \mathcal{E}_{co})$, where the edge e_{ij} indicates that the product v_i and v_j have been interacted with together by the same users. With a large number of connections per product, the original co-interacted product network are quite dense. However, lots of co-occurrence relationships are meaningless, representing spurious associations generated by chance [56]. For example, it is irrational to assert a strong relationship between bread and a memory card based on occasional views from just a few users. Therefore, we propose a noise filtering method to remove unreliable relationships. Formally, we compute the co-occurrence similarity between v_i and v_j as follows:

$$sim_{ij} = \frac{|Set(v_i) \cap Set(v_j)|}{deg(v_i) \cdot \sqrt{|Set(v_i)| \cdot |Set(v_j)|}}, \quad (9)$$

where $Set(v_i)$ denotes the set of the users interacted with the product v_i , $deg(v_i)$ denotes the degree of the product node. v_i and v_j are considered homogeneous if $sim_{ij} < \eta$, where η is the predefined minimum threshold.

Furthermore, the LightGCN [22] is introduced to simulate the propagation of product homophily for learning the co-interacted product features. Formally,

$$z_i^{(l+1)} = \sum_{j \in N_{co}(i)} \frac{1}{\sqrt{|N_{co}(i)| \cdot |N_{co}(j)|}} z_j^{(l)}, \quad (10)$$

$$ci_v = \frac{1}{L} \sum_{l=1}^L z_i^{(l)}, \quad (11)$$

where $N_{co}(i)$ denotes the neighborhood product set of the product v_i .

3.2.3. Product feature fusion layer

The product feature fusion layer combines the collaborative product features cf_v with the co-interacted product features ci_v for learning the comprehensive product representations as follows: $e_i^v = MLP_v[cf_v \parallel ci_v]$, where $MLP_v(\cdot)$ denotes the 3-layer MLP of the form $\mathbb{R}^{2d} \rightarrow \mathbb{R}^d$.

3.3. Prediction and optimization module

After learning the comprehensive features of users and products, the relevance score is calculated to illustrate the extent of the user's interest in the product by the dot-product similarity of their representations. Formally, $\hat{r}_{ij} = (e_i^u \cdot e_j^v) / \tau$, where the temperature hyper-parameter τ is introduced to learn better feature distributions. Moreover, we adopt the joint learning objective function to optimize our DeepSC, benefiting from both the point-wise loss and the pair-wise loss. Formally,

$$\mathcal{L}_{rec} = \lambda \cdot \mathcal{L}_{point} + (1 - \lambda) \mathcal{L}_{pair}, \quad (12)$$

$$\mathcal{L}_{point} = - \sum_{j=1}^{|\mathcal{O}|} r_{ij} \log(\hat{r}_{ij}) + (1 - r_{ij}) \log(1 - \hat{r}_{ij}), \quad (13)$$

$$\mathcal{L}_{pair} = \sum_{(i,j^+,j^-) \in \mathcal{O}} \sigma(\hat{r}_{ij^+} - \hat{r}_{ij^-}), \quad (14)$$

where λ is a hyper-parameter to balance the weights of different losses, r_{ij} denotes the one-hot encoding vector of the ground truth product, and $\mathcal{O}$ denotes the training dataset. The positive and negative samples denote the observed and unobserved user-product interactions, respectively. The point-wise loss focuses on learning a user's absolute interest in a single product, while the pair-wise loss emphasizes learning her relative preference between pairs of products [27]. The proposed joint learning objective function integrates the fitting traits of different losses to achieve precise rating and ranking simultaneously. Additionally, the multi-task learning objective is denoted as follows: $\mathcal{L}_{total} = \mathcal{L}_{rec} + \xi \cdot \mathcal{L}_{aux}$, where ξ is the hyper-parameter to scale the weight of the auxiliary task.

3.4. Analysis of time complexity

The time complexity of the proposed DeepSC framework stems from five key components. First, the time complexity of the random walk-based structural pre-training is $O(|U|\omega\rho(d+1))$, where ω and ρ denote the number of random walks per node and the length of each random walk path, respectively. Second, the time complexity of the two hypergraph attention networks is $O(6|U||V| + |U|^2|V| + |U||V|^2 + 2f|U|^2|V|^2d^2)L$, where f denotes the number of attention heads in the hypergraph attention networks. Third, the time complexity of the gated attention network is $O((|U|d^2 + |\mathcal{E}_s|d)\psi L)$, where ψ and $|\mathcal{E}_s|$ denote the number of attention heads in the gated attention network and the number of social relationships, respectively. Fourth, the time complexity of the LightGCN in the co-interacted feature extractor is $|\mathcal{E}_{co}|Ld$. Lastly, the time complexity of the prediction and optimization module is $O(|V|d)$. Therefore, the overall time complexity of DeepSC is $O((\omega\rho(d+1) + \psi Ld^2)|U| + |V|d + \psi Ld|\mathcal{E}_s| + Ld|\mathcal{E}_{co}| + 6L|U||V| + L|U|^2|V| + L|U||V|^2 + 2fLd^2|U|^2|V|^2)$, where $|U|$, $|V|$, $|\mathcal{E}_s|$, and $|\mathcal{E}_{co}|$ are typically larger than other variables. In practical scenarios, models are usually trained offline using ample computational resources, enabling efficient and timely training. During deployment, inference is optimized for real-time performance. Modern hardware like GPUs further accelerates training, making complex models feasible in real-world applications. Therefore, DeepSC has an acceptable time complexity.

4. Experiments

In this section, we conduct thorough experiments to demonstrate the effectiveness of the proposed DeepSC model by addressing the following questions: (Q1) How does DeepSC compare to 19 competitive baseline models across 3 real-world datasets in social recommendation tasks? (Q2) How does DeepSC perform compared to various representative baselines under different degrees of sparse data? (Q3) What impact do social network and each dimension of the social capital have on enhancing social recommendation performance? (Q4) How does co-interacted product network, hypergraphs, and graph learning methods contribute to improving social recommendation performance? (Q5) How does each part of the objective function influence the model optimization? (Q6) What is the performance variability of DeepSC under different hyperparameter settings?

These six questions are intended to address the two research questions mentioned above from various perspectives. Specifically, Q1 provides insights into the model's overall effectiveness and its advantages over existing methods. Q2 assesses how DeepSC responds to data sparsity to ensure its robustness in practical applications. Q3 explores the role of social capital dimensions — structural, relational, and cognitive — in refining social recommendations. Q4 evaluates the contributions of the key components in improving recommendation accuracy. Q5 examines the impact of different loss terms on model convergence and optimization. Lastly, Q6 investigates the model's sensitivity to hyperparameter choices, offering guidance for optimizing performance across various scenarios.

4.1. Datasets and evaluation metrics

Three popular datasets are used in our experiments: Ciao, Epinions, and Yelp. Specifically, the Ciao dataset is obtained from a popular online shopping websites,⁴ while the Epinions dataset is created by a who-trust-whom online social network of a general consumer review site.⁵ Moreover, the Yelp dataset is sourced from the largest review website for businesses in the United States.⁶ The social networking

⁴ www.ciao.co.uk.

⁵ www.epinions.com.

⁶ www.yelp.com.

Table 1
Statistics of the three datasets.

Dataset	# Users	# Products	# Interactions	# Followers	# Followees	# Social links
Ciao	1925	15 053	33 175	1925	1925	65 084
Epinions	18 081	251 722	746 148	18 081	18 081	572 784
Yelp	99 262	105 142	960 732	99 262	99 262	1 298 522

Table 2

Summary of the baseline models and our DeepSC. S, R, C denotes the structural, relational, and cognitive social capital feature. Pair, Point, and Aux. denotes the pair-wise loss function, the point-wise loss function, and the auxiliary task, respectively.

Model	Social relationships	Social capital	User-product interaction	Product network	w/o side info	Denoising method	Loss function
SAMN [58]	✓	R	Matrix	✗	✓	Soft	Pair
EATNN [59]	✓	None	Knowledge graph	✗	✓	Soft	Point
DiffNet [34]	✓	R	Vanilla bipartite graph	✗	✗	None	Pair
GraphRec [36]	✓	C	Vanilla bipartite graph	✗	✓	Soft	Point
NGCF [60]	✗	None	Vanilla bipartite graph	✗	✓	None	Point
GCCF [61]	✗	None	Vanilla bipartite graph	✗	✓	None	Pair
DGRec [50]	✓	R/C	Interaction sequence	✗	✗	Soft	Point
KGAT [62]	✗	None	Knowledge graph	✗	✗	Soft	Point/Aux.
DGCF [12]	✗	None	Vanilla bipartite graph	✗	✓	Soft	Pair/Aux.
DisenHAN [63]	✗	None	Vanilla bipartite graph	✗	✓	Soft	Pair/Aux.
HAN [64]	✓	R	Heterogeneous graph	✗	✓	Soft	Point
HGT [24]	✓	R	Heterogeneous graph	✗	✓	Soft	Point
HERec [65]	✓	R	Heterogeneous graph	✗	✗	None	Point
MHCN [66]	✓	R	Hypergraph	✗	✓	Soft	Pair/Aux.
DGNN [11]	✓	R	Heterogeneous graph	✗	✗	Soft	Pair
DSL [67]	✓	R	Vanilla bipartite graph	✗	✓	Soft	Pair/Aux.
GDMR [9]	✓	R	Vanilla bipartite graph	✗	✗	Hard	Pair/Aux.
RecDiff [68]	✓	R	Vanilla bipartite graph	✗	✓	Soft	Pair/Aux.
SGSR [69]	✓	R	Vanilla bipartite graph	✗	✓	Soft	Pair/Aux.
DeepSC	✓	S/R/C	Hypergraph	✓	✓	Soft	Point/Pair/Aux.

service on the three online platforms allow users to review and rate businesses, as well as make friends with others. For all the datasets, we follow the practice of [11] and map the rating score to 0 or 1, where the score 1 means the user appreciate the product. Table 1 shows the statistics for the three datasets. Additionally, we select 80% as a training set to learn the parameters, and the rest are divided into a validation set and a testing set on average.

To evaluate the performance of the models, we introduce the widely adopted two metrics, i.e., Hit Rate (HR@K) and Normalized Discounted Cumulative Gain (NDCG@K), where the length of recommendation list K is set as 5, 10, and 20. Specifically, HR@K is the metric that assesses the proportion of test cases in which the correct products are recommended within the top-K list. For each user in the test set, we follow the widely adopted evaluation setting [9,11,57], i.e., randomly sample 100 products that the user has not interacted with and rank them with the positive samples in the test set.

4.2. Baseline models and implementation settings

To demonstrate the superiority of our DeepSC, we compare it with four groups of 19 representative baseline models as shown in Table 2. Our DeepSC is implemented based on the popular recommendation framework RecBole⁷ and its extension RecBole-GNN⁸ for easy development and reproduction. The embedding dimension is searched from {8, 16, 32, 64, 128}. The batch size is searched from {64, 128, 256, 512, 1024}. The stacking layer of graph neural networks is searched from {1, 2, 3, 4}. The number of attention heads is searched from {1, 2, 4, 8}. The negative sampling ratio and the dropout ratio are set to 0.2. The temperature parameter is set to 0.07. The AdamW optimizer [70] is used to train parameters with the learning rate of 0.001. Early stopping strategy is adopted to alleviate over-fitting, i.e., the training is stopped when NDCG@10 on the validation set is not promoted for 5 consecutive epochs. The implementation of our model can be found at <https://github.com/usernameAI/DeepSC>.

⁷ <https://github.com/RUCAIBox/RecBole>.

⁸ <https://github.com/RUCAIBox/RecBole-GNN>.

4.3. Overall performance

Tables 3 and 4 show the HR@K and NDCG@K results on the three datasets. To address Q1, we have the following findings.

(1) Our DeepSC achieves better performance than the baseline models disregarding the effect of social relationships [12,60–63], demonstrating that user preferences tend to be influenced by their social friends. It is attributed that people frequently seek advice from friends in their social networks before purchasing a product or using a service. Therefore, extracting hidden social preferences is crucial for enhancing the performance of recommender systems.

(2) Compared to the baseline models considering just one or two aspects of social capital [34,36,50], our DeepSC shows significant improvements in recommendation performance. This observation suggests that integrating multiple dimensions of social capital enhance the assessment of social relationship robustness, which helps in learning better representations of user preferences from social networks.

(3) Our DeepSC outperforms the baseline models disregarding the product network [67–69], showing that products do not exist in isolation but rather influence each other. Especially, our co-interacted network, as a unique form of product network, demonstrates how products influence interactions with other products. It is noted that some baseline models [9,11,65] introduce external side information to improve the feature learning of products. Despite some advantages, the selection of appropriate side information depends on extensive domain knowledge, struggling to generalize to other scenarios. Moreover, these excessive side information reduced the efficiency of models, hindering the large-scale deployment in real scenarios.

(4) Compared to the baseline model merely based on point-wise or pair-wise objective function [65,66], our DeepSC combine both of them to improve the calibration ability and ranking ability simultaneously. Although widely adopted the pair-wise objective functions emphasize

Table 3
Top-K performance comparison on HR metric.

Model	Ciao			Epinions			Yelp		
	@5	@10	@20	@5	@10	@20	@5	@10	@20
SAMN	0.3468	0.4677	0.6251	0.5176	0.6390	0.7491	0.6359	0.7971	0.9009
EATNN	0.2969	0.4130	0.5222	0.5283	0.6422	0.7501	0.6425	0.7273	0.8066
DiffNet	0.3941	0.5202	0.6647	0.5106	0.6323	0.7367	0.6701	0.8222	0.9053
GraphRec	0.3058	0.4594	0.5976	0.5683	0.6865	0.8001	0.6631	0.8019	0.8944
NGCF	0.3570	0.4843	0.5937	0.5612	0.6944	0.8010	0.6748	0.8204	0.9011
GCCF	0.3685	0.4926	0.6289	0.5538	0.6779	0.7906	0.6703	0.8130	0.9011
DGRec	0.3724	0.5086	0.6219	0.5053	0.6268	0.7308	0.6511	0.7830	0.8824
KGAT	0.3391	0.4907	0.6052	0.5483	0.6756	0.7880	0.6503	0.7737	0.8795
DGCF	0.3871	0.5189	0.6775	0.5479	0.6635	0.7770	0.6565	0.7956	0.9010
DisenHAN	0.3493	0.4856	0.6161	0.5609	0.6825	0.7890	0.6511	0.8159	0.9040
HAN	0.2937	0.4856	0.6513	0.5403	0.6673	0.7761	0.6635	0.8169	0.8977
HGT	0.3415	0.4933	0.6128	0.5757	0.7001	0.8053	0.6888	0.8185	0.9060
HERec	0.3832	0.5298	0.6846	0.5519	0.6767	0.7792	0.5833	0.7047	0.8125
MHCN	0.3864	0.5080	0.6321	0.5199	0.6411	0.7496	0.6607	0.8019	0.8958
DGNN	<u>0.4120</u>	<u>0.5515</u>	<u>0.6942</u>	<u>0.6142</u>	<u>0.7335</u>	<u>0.8281</u>	<u>0.7052</u>	<u>0.8373</u>	0.9293
DSL	0.1280	0.1919	0.2559	0.0675	0.1289	0.2271	0.0536	0.1205	0.2153
GDMSR	0.1959	0.2678	0.3504	0.3450	0.4120	0.5091	0.5306	0.6839	0.8304
RecDiff	0.0781	0.0938	0.2184	0.0563	0.1077	0.2034	0.0582	0.1034	0.2023
SGSR	0.1122	0.1869	0.2960	0.0931	0.1585	0.2474	0.2288	0.3347	0.4473
DeepSC	0.5074	0.6675	0.8240	0.7692	0.8613	0.9317	0.7342	0.8438	<u>0.9252</u>
Improve.	23.16%***	21.03%***	18.70%***	25.24%***	17.42%**	12.51%***	4.11%**	0.78%**	-0.44%*

¹ For each dataset, the **bold**-faced number is the best score and the second performer is underlined.

* Denote the level of marginal significance at the 0.05 level.

** Denote the level of marginal significance at the 0.01 level.

*** Denote the level of marginal significance at the 0.001 level.

Table 4
Top-K performance comparison on NDCG metric.

Model	Ciao			Epinions			Yelp		
	@5	@10	@20	@5	@10	@20	@5	@10	@20
SAMN	0.2460	0.2838	0.3223	0.3860	0.4259	0.4553	0.4662	0.5293	0.5407
EATNN	0.2124	0.2520	0.2819	0.3924	0.4483	0.4557	0.4866	0.5289	0.5468
DiffNet	0.2816	0.3201	0.3573	0.3820	0.4160	0.4476	0.5127	0.5524	0.5701
GraphRec	0.2235	0.2670	0.3042	0.4325	0.4786	0.5011	0.4903	0.5372	0.5650
NGCF	0.2360	0.3088	0.3188	0.4316	0.4763	0.5006	0.5192	0.5651	0.5684
GCCF	0.2668	0.3070	0.3414	0.4161	0.4783	0.4852	0.5130	0.5585	0.5503
DGRec	0.2647	0.3113	0.3277	0.3775	0.4127	0.4429	0.4897	0.5386	0.5611
KGAT	0.2422	0.2977	0.3326	0.4139	0.4708	0.4837	0.4901	0.5386	0.5521
DGCF	0.2782	0.3166	0.3604	0.4144	0.4594	0.4811	0.4958	0.5410	0.5678
DisenHAN	0.2482	0.2894	0.3248	0.4247	0.4627	0.4911	0.4944	0.5403	0.5650
HAN	0.1897	0.2608	0.2821	0.4106	0.4371	0.4802	0.5080	0.5511	0.5529
HGT	0.2372	0.3062	0.3229	0.4360	0.4812	0.5029	0.5136	0.5547	0.5802
HERec	0.2679	0.3104	0.3641	0.4179	0.4572	0.4839	0.4501	0.4990	0.5034
MHCN	0.2799	0.3118	0.3453	0.3883	0.4261	0.4551	0.4911	0.5348	0.5670
DGNN	0.2890	0.3338	<u>0.3726</u>	0.4794	0.5215	0.5387	0.5378	0.5873	0.6043
DSL	0.1043	0.1271	0.1417	0.0406	0.0594	0.0829	0.0346	0.0562	0.0800
GDMSR	0.2876	0.3206	0.3487	<u>0.5328</u>	<u>0.5687</u>	<u>0.5939</u>	<u>0.5460</u>	<u>0.5949</u>	<u>0.6170</u>
RecDiff	0.0412	0.0421	0.0763	0.0332	0.0433	0.0724	0.0333	0.0429	0.0720
SGSR	0.2431	0.2465	0.2729	0.3538	0.3000	0.3318	0.2957	0.3232	0.3587
DeepSC	0.3220	0.3773	0.4246	0.5374	0.5754	0.6069	0.5572	0.5968	0.6218
Improve.	11.42%**	13.03%***	13.96%***	0.86%***	1.18%**	2.19%***	2.05%***	0.32%*	0.78%**

¹ For each dataset, the **bold**-faced number is the best score and the second performer is underlined.

* Denote the level of marginal significance at the 0.05 level.

** Denote the level of marginal significance at the 0.01 level.

*** Denote the level of marginal significance at the 0.001 level.

the relative preference on a pair of products, they fail to estimate the click probability on the specific product, leading to sub-optimal performance. Moreover, our social link prediction task is introduced to align the preferences between the focal user and her social friend, which helps in better assigning attention weights to each social friend.

(5) Our DeepSC achieves better performance than the baseline models without denoising method or adopting hard denoising method [9, 65], suggesting that the soft denoising methods are more suitable to capture social influence. Instead of simply filtering out social friends with low similarity, the soft denoising methods adaptively diminish the impact of heterogeneous social friends while retaining more valuable social opinions. Moreover, our gated attention network combines two

attention mechanism to gather unbiased social opinions, achieving better denoising capability compared to existing methods.

(6) The hypergraph-based models [66] generally achieve promising performance, indicating that hypergraphs better represent the user-product interactions. Except the vanilla bipartite graph, recent works adopt various knowledge graphs and heterogeneous graphs to represent the relations among users, products, and various contextual features. However, the sparse node types and edge types present in these graphs make it challenging to learn effective representations. Furthermore, we introduce the attention mechanism to enhance the traditional hypergraph convolution [66] by learning a dynamic transition matrix that enables an adaptive information propagation between product nodes.

Table 5
Statistics of the five grouping datasets.

Dataset	# Users	# Products	# Interactions	Avg. inter. per user	Sparsity
(0, 20]	90 049	83 499	538 632	5.98	99.9928%
(20, 40]	5675	51 140	160 698	28.32	99.9446%
(40, 80]	2628	47 935	145 295	55.29	99.8847%
(80, 160]	763	36 556	80 956	106.10	99.7097%
(160,)	147	22 661	35 151	239.12	98.9448%

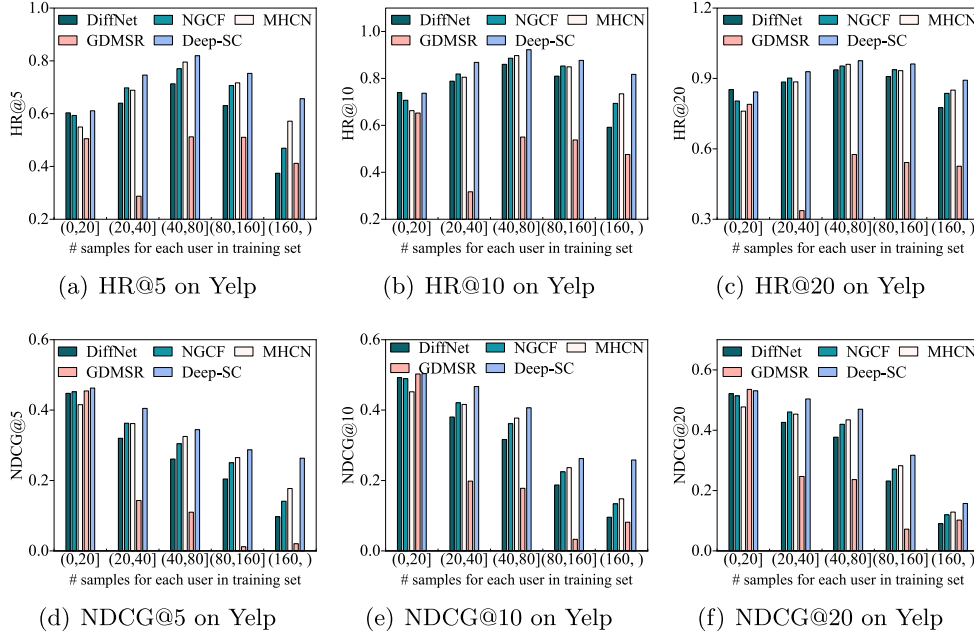


Fig. 3. Performance comparison for different user groups.

4.4. User group study

To address Q2, we evaluate the recommendation performance across various interaction sparsity levels to verify the advantages of DeepSC in tackling the data sparsity problem. Following [71], we categorize users in the Yelp dataset into five groups based on their number of interactions. For example, the interval (0,20] represents the users who have at least one interaction and less than 21 interactions. The statistics of the five sub-datasets is shown in Table 5. Subsequently, we compare DeepSC with five representative baseline models and present the results in Fig. 3. It is observed that as the number of interactions continues to increase, the performance on the HR metric first rises and then falls, while the NDCG metric consistently declines. Despite the increasing density of interaction data, the total number of users in each sub-dataset is decreasing. The reduction in user count results in fewer training samples for the model, limiting its ability to learn representative interaction patterns effectively. Moreover, we find that our DeepSC consistently outperforms all the baseline models, no matter for cold-start or active user groups. On one hand, when the interaction data is too sparse to effectively capture a user's preferences, DeepSC leverages opinions from their social friends to infer potential preferences. On the other hand, even when there are many interactions, DeepSC captures the primary intentions by aligning the user's preferences with those of their social friends.

4.5. Ablation study

To address Q3 and Q4, we compare DeepSC with five variant models to demonstrate the effectiveness of our design. Each variant model is introduced as follows: (1) w/o social network, which views the cognitive social capital feature as user preference and retains the whole product

feature learning module; (2) w/o structural feature, which removes the structural social capital feature to capture user's social preference; (3) w/o relational feature, which removes the relational social capital feature to capture user's social preference; (4) w/o cognitive feature, which removes the relational social capital feature to capture user's social preference; (5) w/o co-interacted network, which disregards the product features hidden in the co-interacted network; (6) w/o user-side hypergraph, which replaces the hypergraph with the bipartite interaction graph for cognitive feature learning; (7) w/o all hypergraphs, which replaces the user-side and product-side hypergraphs with the bipartite interaction graph; (8) Unified heterogeneous graph, which adopts a unified heterogeneous graph incorporates interaction relationships, social relationships, and co-occurrence relationships to facilitate user and product feature learning; (9) Variant of attention, which replaces the attention weight calculation in Eq. (5) with the following formulation: $\alpha_{i,j} = \text{Softmax} \left(\omega \left(q_s^T \left[W_3 h_i^u || W_3 h_j^u \right] \cdot \left((W_4 h_i^u)^T W_4 h_j^u \right) \right) \right)$; (10) w/o hypergraph attention, which replaces attention network with convolutional network in hypergraphs.

Based on the experimental results as shown in Table 6, our DeepSC consistently outperforms all the variant models, achieving a performance gain of up to 13.96%, which highlights the effectiveness of our model design. Specifically, compared to the variant w/o social network, it is evident that the homophily and social influence extractor contributes most significantly to performance improvement, especially when the social network is denser. This observation confirms the roles of homophily and social influence in recommendations, where users tend to seek advice from friends who are similar to themselves. Furthermore, DeepSC outperforms its variant that ignores any dimensional feature of social capital, demonstrating that each social capital dimension contributes uniquely to social preference learning. These observations confirm our research motivation that capturing multiple

Table 6
Ablation study results.

Dataset	Model	HR@5	NDCG@5	HR@10	NDCG@10	HR@20	NDCG@20
Ciao	w/o social network	0.4006	0.2535	0.5143	0.2895	0.6533	0.3264
	w/o structural feature	0.4033	0.2577	0.5164	0.2951	0.6432	0.3297
	w/o relational feature	0.4979	0.3113	0.6385	0.3579	0.7770	0.3999
	w/o cognitive feature	0.4937	0.3020	0.6432	0.3516	0.7865	0.3938
	w/o co-interacted network	0.3377	0.2163	0.4212	0.2439	0.5259	0.2716
	w/o user-side hypergraph	0.4630	0.2911	0.6073	0.3379	0.7553	0.3808
	w/o all hypergraphs	0.4678	0.2869	0.6165	0.3353	0.7719	0.3809
	Unified heterogeneous graph	0.3097	0.1953	0.4043	0.2248	0.5259	0.2556
	Variant of gated attention	0.4535	0.2895	0.5904	0.3362	0.7563	0.3828
	w/o hypergraph attention	0.3922	0.2440	0.5053	0.2800	0.6295	0.3133
	Deep-SC	0.5074	0.3220	0.6675	0.3773	0.8240	0.4246
Epinions	w/o social network	0.7034	0.4963	0.7752	0.5226	0.8354	0.5444
	w/o structural feature	0.7200	0.5116	0.7965	0.5387	0.8537	0.5611
	w/o relational feature	0.7515	0.5365	0.8301	0.5658	0.8863	0.5889
	w/o cognitive feature	0.7492	0.5337	0.8269	0.5624	0.8864	0.5858
	w/o co-interacted network	0.7241	0.5180	0.7938	0.5432	0.8482	0.5644
	w/o user-side hypergraph	0.7239	0.5180	0.8004	0.5468	0.8624	0.5694
	w/o all hypergraphs	0.7200	0.5113	0.7883	0.5365	0.8432	0.5574
	Unified heterogeneous graph	0.6956	0.4935	0.7677	0.5188	0.8261	0.5399
	Variant of gated attention	0.7409	0.5224	0.8163	0.5512	0.8761	0.5753
	w/o hypergraph attention	0.7135	0.5109	0.7859	0.5365	0.8450	0.5587
	Deep-SC	0.7692	0.5374	0.8613	0.5754	0.9317	0.6069
Yelp	w/o social network	0.7120	0.5410	0.8170	0.5789	0.8945	0.6024
	w/o structural feature	0.7307	0.5510	0.8428	0.5912	0.9234	0.6161
	w/o relational feature	0.7139	0.5406	0.8245	0.5800	0.9047	0.6045
	w/o cognitive feature	0.7337	0.5551	0.8430	0.5943	0.9220	0.6186
	w/o co-interacted network	0.7012	0.5246	0.8125	0.5641	0.8956	0.5893
	w/o user-side hypergraph	0.7275	0.5548	0.8352	0.5937	0.9179	0.6188
	w/o all hypergraphs	0.7158	0.5327	0.8365	0.5757	0.9307	0.6042
	Unified heterogeneous graph	0.7030	0.5342	0.8094	0.5719	0.8917	0.5967
	Variant of gated attention	0.7273	0.5528	0.8385	0.5927	0.9206	0.6178
	w/o hypergraph attention	0.7264	0.5569	0.8303	0.5941	0.9064	0.6175
	Deep-SC	0.7342	0.5572	0.8438	0.5968	0.9252	0.6218

¹ For each dataset, the **bold-faced** number is the best score.

user preferences facilitates more accurate recommendations. Notably, structural social capital is most critical on denser datasets like Ciao and Epinions, while relational features are most influential on the sparser dataset, Yelp. One reason is that a sparser social network provides insufficient information for effective structural feature learning, causing the model to rely more on the relational feature for recommendations.

In the following four ablations, we investigate the effectiveness of the co-interacted network and the two hypergraphs. Specifically, removing the co-interacted network leads to lower performance. This aligns with empirical findings [72] that products do not exist in isolation but influence each other's sales. The co-interacted network offers richer internal relations for improved product feature learning, maintaining high efficiency compared to incorporating external information like category and price. Moreover, adopting the traditional bipartite interaction graph for user-side cognitive feature learning and product-side collaborative feature learning fails to gain more performance improvement, demonstrating the superiority of hypergraphs. This can be attributed to hypergraphs being more effective for representing group-level relations, such as the focal product viewed by a group of users and a user's market basket. Furthermore, we introduce a unified heterogeneous graph for user and product feature learning. However, this variant fails to achieve promising performance, indicating that the model struggles to learn representative features from the complex relationships in naive heterogeneous graph. It is more rational to tailor appropriate graphs that effectively reflect the relations among entities to enhance performance.

In the last two ablations, we investigate the effectiveness of the feature learning methods. Specifically, the gated attention network outperforms its variant, showing its effectiveness on social preference learning. It can be attributed to the dot-product attention with the sigmoid function represents the reliability of social relations, allowing it to softly filter out unreliable friends, while implicitly highlighting the remaining reliable friends. Moreover, compared to the variant

w/o hypergraph attention, our DeepSC achieves better performance, suggesting the effectiveness of the attention mechanism adopted in hypergraph learning. Hypergraph attention network effectively captures the interaction relationships between users and products by learning a dynamic incidence matrix.

4.6. Effect of the multi-task learning objective function

To address Q5, we adjust the loss weights λ and ξ within $\{1, 0.1, 0.3, 0.5, 0.7, 1\}$ and $\{0, 0.001, 0.001, 0.01, 0.1\}$, respectively. Specially, the joint learning objective function degrades into the pair-wise objective function when $\lambda = 0$, while degrading into the point-wise objective function when $\lambda = 1$. Moreover, we remove the social link prediction auxiliary task when sets $\xi = 0$. The experimental results are shown in Fig. 4. We have the following observations.

First, our DeepSC consistently achieve the sub-optimal performance when sets $\lambda = 0$ or 1, demonstrating the effectiveness of combining the point-wise objective function with the pair-wise objective function. Specifically, the point-wise objective function facilitates the specific preference learning on each product, while the pair-wise objective function focuses on learning the relative preference on each pair of products. By contrast, our joint learning objective function benefits from both point-wise and pair-wise objective function, improving the ability of click-through-rate estimation and ranking simultaneously.

Second, our DeepSC is not comparable when sets $\xi = 0$, suggesting that the social link prediction auxiliary task effectively enhances the social preference learning by aligning the preference representations between the focal user and each social friend. Moreover, DeepSC achieves the best performance on Epinions and Yelp with $\xi = 0.1$, while the optimal performance is observed at $\xi = 0.01$ for the Ciao dataset. Therefore, it is necessary to select an appropriate ξ value when introducing the auxiliary task to train DeepSC.

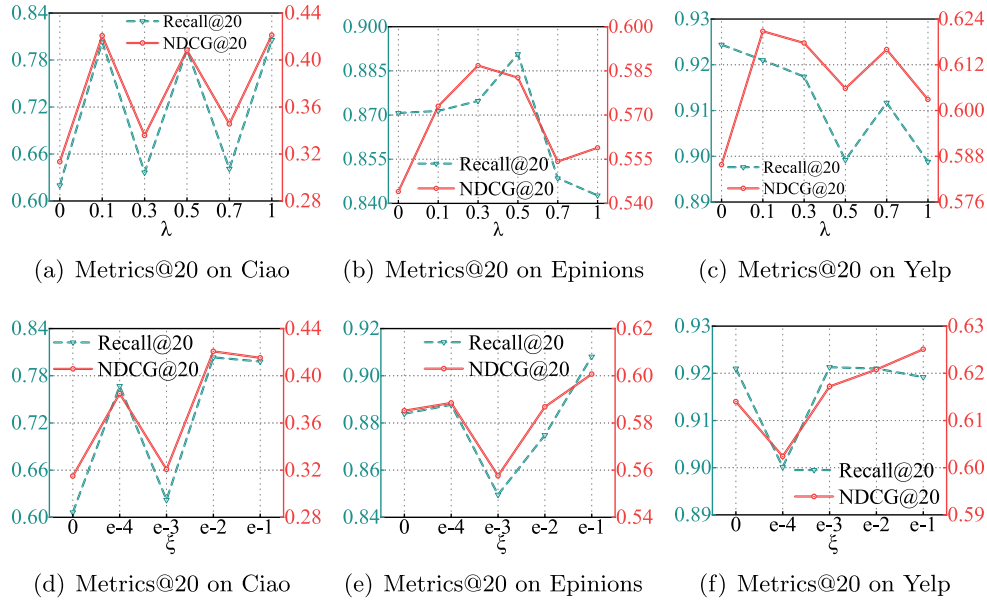


Fig. 4. DeepSC performance with varying loss weights on the three datasets.

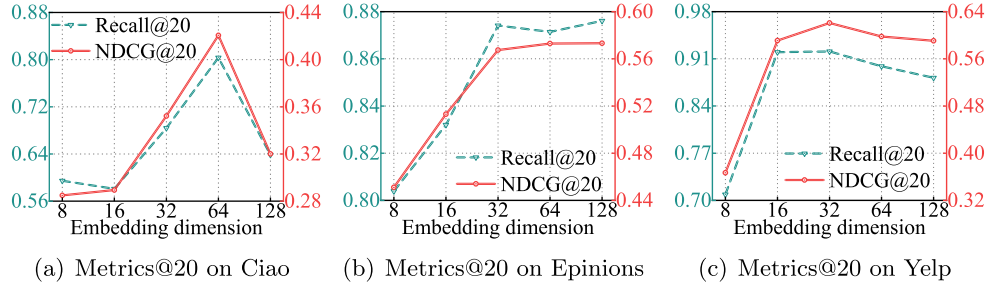


Fig. 5. DeepSC performance with varying embedding dimensions on the three datasets.

4.7. Parameter sensitivity analysis

To address Q6, we investigate the impact of the main hyper-parameter settings, including embedding dimension, the number of stacking layers, and the number of attention heads. Specifically, we report the experiment results of different embedding dimension ranging in $\{8, 16, 32, 64, 128\}$. As shown in Fig. 5, it is observed that the embedding dimension of 32 or 64 is sufficient to achieve promising performance on the three datasets. The dimension of embedding controls the representation learning capacity of the model. A smaller embedding dimension struggles to provide the sufficient large feature space for representation learning, while a larger embedding dimension increases the risk of over-fitting. Compared to the Ciao dataset, the Epinions and Yelp datasets contains much more users, products, interactions, and social relationships. Therefore, DeepSC benefits more from larger embedding dimension on the two datasets.

As shown in Fig. 6, we tune the number of stacking layers in $\{1, 2, 3, 4\}$. It is found that our model experiences performance degradation with an increase in the number of stacking layers. One reason is that the well-designed hyper-graphs and co-interacted network significantly reduce the difficulty of DeepSC in learning promising user and product features. Specifically, the hyper-graphs represents the relations involved multiple users and products, while the co-interacted network explicitly contains the high-order relations among products. Moreover, the social link prediction auxiliary task effectively enhances the capacity of social preference learning. Generally speaking, setting one layer is sufficient for DeepSC to learn satisfactory features, which also helps reduce the training cost.

The similar trends can be observed in Fig. 7, where the number of attention head is adjusted at $\{1, 2, 4, 8\}$ on the three datasets. In the denser datasets, i.e. Ciao and Epinions, configuring our model with just one attention head yields promising performance, while more attention heads lead to a decreasing tendency. By contrast, in the sparser Yelp dataset, using four to eight attention heads can lead to improved performance on Recall@20. One reason is that the single attention mechanism struggles to capture robust user preferences in sparse user-product interactions and social network, while multiple attention heads allow more comprehensive preference learning from different representation subspaces.

4.8. In-depth analysis with visualization

In this section, we conduct two case studies to demonstrate how our DeepSC works. Specifically, in the first case study, based on the results from Section 4.7, we implement our DeepSC model with four hyper-graph attention heads on the Yelp dataset, as shown in Fig. 8. After that, we visualize the attention map of each head and the correlations of each pair of attention heads, as shown in Fig. 9. We observe that these attention maps differ from each other, and the pairwise correlation values are low. This suggests that, although all attention heads receive the same input features, they tend to specialize in different aspects of user preferences during training. Such specialization enables a more fine-grained modeling of user intent. This analysis not only validates the effectiveness of the multi-head attention mechanism in capturing the complexity of user behavior but also improves the explainability of

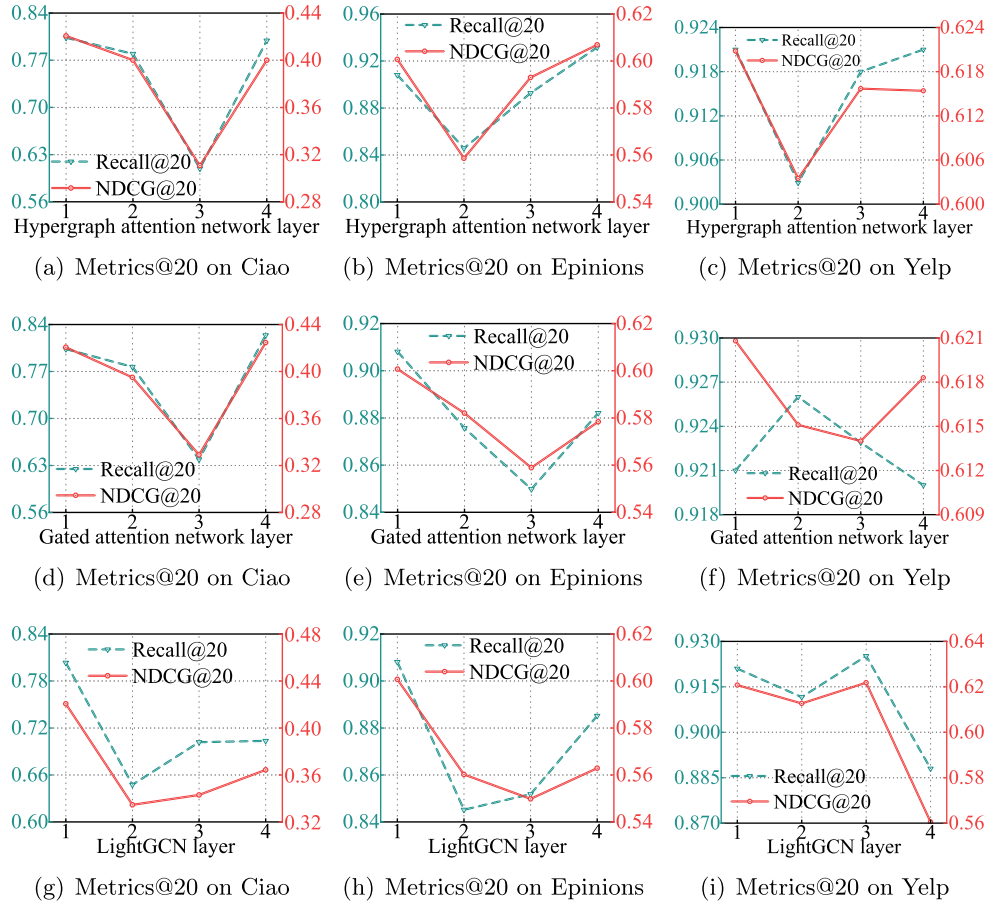


Fig. 6. DeepSC performance with varying stacking layers on the three datasets.

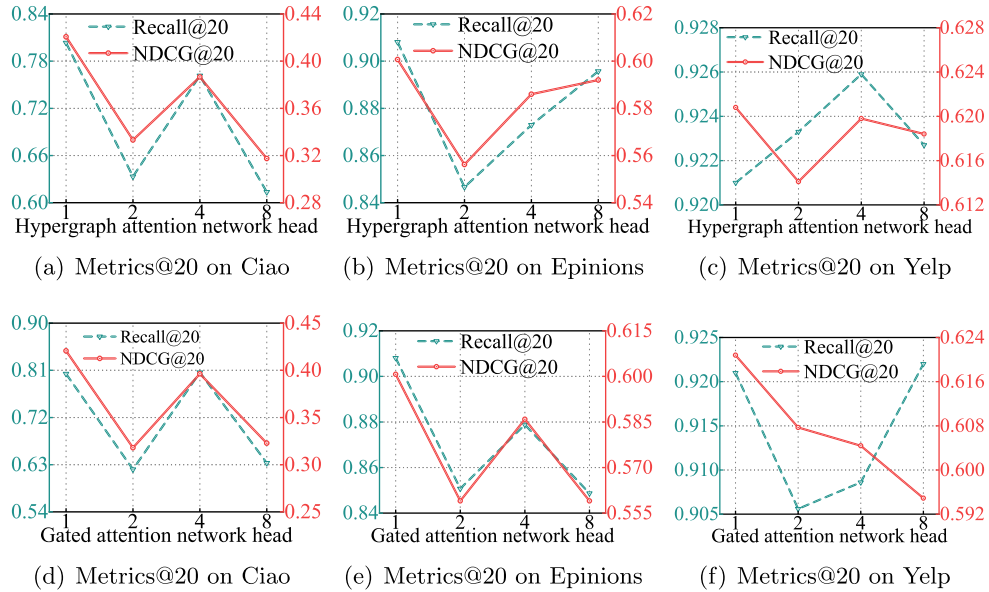


Fig. 7. DeepSC performance with varying attention heads on the three datasets.

the model. It provides intuitive insights into how users make decisions under social influence from multiple perspectives.

In the second case study, our goal is to examine how the three social capital dimensional features affect the recommendation list. We first implement our DeepSC model and three variants, each with one of the features removed, respectively, on the Ciao dataset. Then, we randomly

select a user and compare the top-5 recommendation lists generated by the four models, along with the predicted interaction probabilities for each recommended product. As shown in Fig. 9, removing each dimensional feature leads to noticeable changes in recommendation list, highlighting the unique roles that these features play in shaping personalized recommendations. Specifically, when the structural

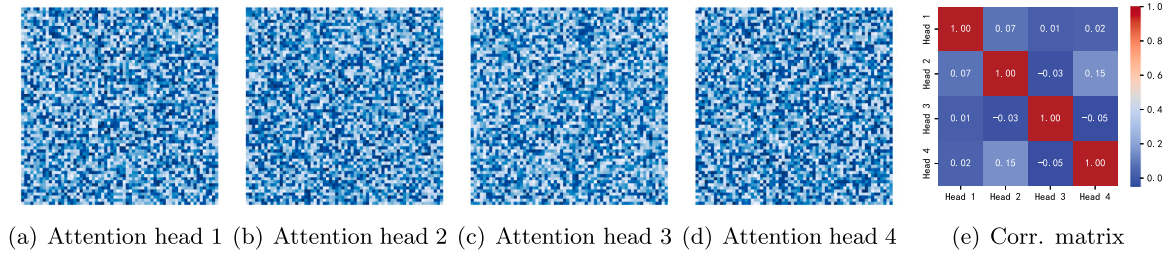


Fig. 8. Visualized attention maps and the correlation matrix.

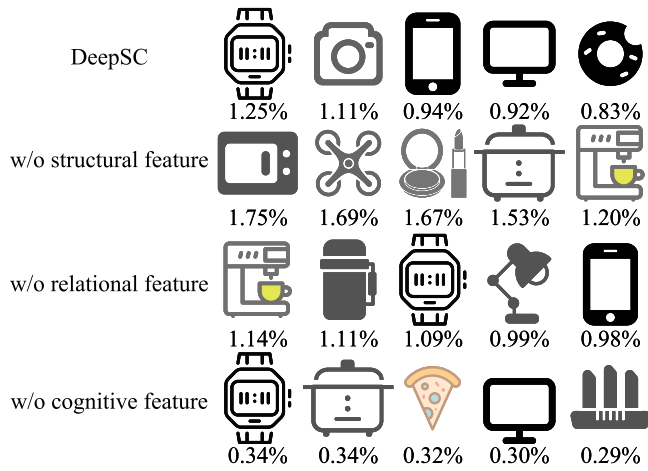


Fig. 9. Top-5 recommendation lists and predicted interaction probabilities of our DeepSC and three variants.

feature is removed, the model tends to overlook products preferred by social friends, such as the digital products. Additionally, removing the relational feature results in poor inference of latent preferences, such as food choices, demonstrating its importance in understanding underlying user tastes. Furthermore, although the recommendation list of the variant without the cognitive feature is most similar to that of the full DeepSC model, the predicted interaction probabilities differ significantly between the two models, such as digital watch and television. This highlights the importance of reinforcing hidden preferences derived from historical interactions for more accurate predictions.

5. Discussion

In this paper, we design a social capital theory-driven social recommender system. Our DeepSC framework incorporates multi-dimensional social capital features to capture diverse social preferences. Additionally, we introduce a co-interacted network to represent the potential relations among products. Furthermore, we propose a multi-task learning approach to optimize model training, including the joint learning task and the auxiliary task. The experimental results suggest that DeepSC outperforms the representative baseline models. Moreover, the in-depth analyses verifies the effectiveness of each component and the robustness of our model. The findings of the study have several research contributions and practical implications that are worthy of discussion.

5.1. Research contributions

This research makes four-fold contributions. First, our DeepSC extracts multiple user preferences guided by the social capital theory. Previous social recommender systems merely capture a single user preference, while few attempts adopt disentangled learning and cluster-based methods to capture potential various preferences, leading to sub-optimal performance and lower explainability. By contrast, our DeepSC

parametrizes all the three dimensions of the social capital to represent multiple user preferences, providing insights for future theory-driven model design. The experimental results demonstrate that our theory-driven design achieves better performance than other intuition-driven designs.

Second, this research expands the theoretical foundation of social recommender systems by introducing social capital theory. Most works are based on the theories of homogeneity and social influence to seek suggestions from social friends [1,73], disregarding the essential role of social capital in explainable and effective social preference learning [74,75]. Our DeepSC provides a solid foundation for the future development of social recommender systems that aim to capture more comprehensive user preferences.

Third, we tailor some components to enhance the social capital-aware feature learning, such as the hypergraph attention networks on interaction hyper-graphs and the LightGCN on co-interacted network. Compared to the vanilla bipartite graph-based methods [9,68,69], the hyper-graphs provides advantages in capturing the set-level interaction relationships. Furthermore, we adopt attention mechanism to better reveal the intrinsic relationship between hyper-nodes by learning a dynamic incidence matrix. Compared to the traditional hyper-graph convolution network [18,66], the hypergraph attention network has stronger feature learning ability. In addition, compared to the methods [11,65] introduce various external side information and adopt heterogeneous graph to manage multiple entities, our approach simplifies access to information while maintaining a lower memory cost. Subsequently, we adopt the parameter-efficient LightGCN to capture the product features based on homogeneous relationships, complementing the heterogeneous product features derived from user-product interactions.

Finally, we propose the multi-task objective learning task to better optimize our model training, consisting of the joint-learning task and the auxiliary task. For the joint-learning task, existing social recommender systems employ cross-entropy-based point-wise loss function and Bayesian personalized ranking-based pair-wise loss function [18,76], disregarding the benefits of fusing both loss functions. Our DeepSC takes the linear approach to combine point-wise with pair-wise loss function, performing well on click-through-rate estimation and ranking simultaneously. To the best of our knowledge, this research is the first to explore the effectiveness of combining point-wise and pair-wise loss functions for social recommendation. It provides both technical and empirical insights into ways to enhance social recommendation systems.

5.2. Practical implications

In practice, our DeepSC has the potential to benefit multiple stakeholders, including users, product providers, and online platform owners, resulting in advantages for all parties involved. The superior performance of our method demonstrates its ability to accurately capture users' multiple preferences and support online decision-making, saving users considerable time in finding products of interest. For product providers, DeepSC effectively facilitates sales by uncovering hidden

relationships among products. This insight allows providers to adjust stock levels based on the recommendations generated by DeepSC. For online platform owners, DeepSC aids in identifying products that users are more likely to engage with, allowing for targeted promotions that enhance user satisfaction and retention. Moreover, deploying more effective recommendation methods can significantly help platform owners increase profits by driving higher conversion rates. Finally, our DeepSC is not limited to deployment in online shopping platforms. It can also serve as a foundational model for various other scenarios due to its broad applicability, such as location-based social recommendation and social-aware music recommendation.

5.3. Limitations and future work

Despite the promising results achieved by our model, several limitations suggest directions for future research. Specifically, DeepSC does not incorporate any side information, which could potentially provide a more comprehensive understanding of user preferences to enhance recommendation precision. In future work, we aim to extend the framework by integrating various types of side information, such as product attributes, product photos, and textual reviews. By leveraging these additional side information, the feature learning capacity is expected to be improved. Moreover, we observe that social networks often exhibit community structures, which are not explicitly utilized in DeepSC. As a promising future research direction, we plan to incorporate social community detection techniques [77] into our framework. This extension has its potential to open new avenues for exploring the interplay between community dynamics and recommendation systems.

CRedit authorship contribution statement

Weiyue Li: Writing – original draft, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ming Gao:** Writing – review & editing, Visualization, Validation, Supervision, Resources, Funding acquisition. **Bowei Chen:** Writing – review & editing. **Jingmin An:** Writing – review & editing. **Yeming Gong:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported and funded by the National Natural Science Foundation of China (No. 72293563, No. 72501053, 72442025); the Natural Science Foundation of Liaoning Province (No. 2024-MS-175); the Basic Scientific Research Project of Liaoning Provincial Department of Education (No. JYTZD2023050); the Liaoning Province Key Research and Development Project (No. 2024JH2/102400020); and the Dalian Scientific and Technological Talents Innovation Support Plan (No. 2022RG17). The authors would like to extend sincere gratitude to Zilong Liu, Maoxin (Molson) Han, Hao Dong, and Cheng Chen from the School of Management Science and Engineering at Dongbei University of Finance and Economics for their valuable discussions and insightful comments on this work. We also wish to thank Hongyu Wang of the Institute of Computing Technology, Chinese Academy of Sciences for his technical support and expertise.

Data availability

Data will be made available on request.

References

- [1] Y. Li, C. Wu, C. Lai, A social recommender mechanism for e-commerce: Combining similarity, trust, and relationship, *Decis. Support Syst.* 55 (3) (2013) 740–752.
- [2] M. Zhang, Y. Liu, A commentary of TikTok recommendation algorithms in MIT technology review 2021, *Fundam. Res.* 1 (6) (2021) 846–847.
- [3] C. Eksombatchai, P. Jindal, J.Z. Liu, Y. Liu, R. Sharma, C. Sugnet, M. Ulrich, J. Leskovec, Pixie: A system for recommending 3+ billion items to 200+ million users in real-time, in: *Proceedings of the WWW*, 2018, pp. 1775–1784.
- [4] X. Li, L. Sun, M. Ling, Y. Peng, A survey of graph neural network based recommendation in social networks, *Neurocomputing* 549 (2023) 126441.
- [5] O. Arazy, N. Kumar, B. Shapira, A theory-driven design framework for social recommender systems, *J. Assoc. Inf. Syst.* 11 (9) (2010) 2.
- [6] L. Ma, R. Krishnan, A.L. Montgomery, Latent homophily or social influence? An empirical analysis of purchase within a social network, *Manag. Sci.* 61 (2) (2015) 454–473.
- [7] G. Guo, J. Zhang, N. Yorke-Smith, TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings, in: *Proceedings of the AAAI*, 2015, pp. 123–129.
- [8] M. Jamali, M. Ester, *TrustWalker*: a random walk model for combining trust-based and item-based recommendation, in: *Proceedings of the KDD*, 2009, pp. 397–406.
- [9] Y. Quan, J. Ding, C. Gao, L. Yi, D. Jin, Y. Li, Robust preference-guided denoising for graph based social recommendation, in: *Proceedings of the WWW*, 2023, pp. 1097–1108.
- [10] W. Ma, Y. Wang, Y. Zhu, Z. Wang, M. Jing, X. Zhao, J. Yu, F. Tang, MADM: a model-agnostic denoising module for graph-based social recommendation, in: *Proceedings of the WSDM*, 2024, pp. 501–509.
- [11] L. Xia, Y. Shao, C. Huang, Y. Xu, H. Xu, J. Pei, Disentangled graph social recommendation, in: *Proceedings of the ICDE*, 2023, pp. 2332–2344.
- [12] X. Wang, H. Jin, A. Zhang, X. He, T. Xu, T. Chua, Disentangled graph collaborative filtering, in: *Proceedings of the SIGIR*, 2020, pp. 1001–1010.
- [13] X. Du, H. Liu, L. Jing, Additive co-clustering with social influence for recommendation, in: *Proceedings of the RecSys*, 2017, pp. 193–200.
- [14] L. Wang, X. Zhao, N. Liu, Z. Shen, C. Zou, Cognitive process-driven model design: A deep learning recommendation model with textual review and context, *Decis. Support Syst.* 176 (2024) 114062.
- [15] C. Perez, I. Ting, Can you hold an advantageous network position? The role of neighborhood similarity in the sustainability of structural holes in social networks, *Decis. Support Syst.* 158 (2022) 113783.
- [16] P.R. de Souza, F.A. Durão, Exploiting social capital for improving personalized recommendations in online social networks, *Expert Syst. Appl.* 246 (2024) 123098.
- [17] J. Nahapiet, S. Ghoshal, Social capital, intellectual capital, and the organizational advantage, *Acad. Manag. Rev.* 23 (2) (1998) 242–266.
- [18] L. Yu, W. Gong, D. Zhang, Live streaming channel recommendation based on viewers' interaction behavior: A hypergraph approach, *Decis. Support Syst.* 184 (2024) 114272.
- [19] D. Kim, A. Oh, How to find your friendly neighborhood: Graph attention design with self-supervision, in: *Proceedings of the ICLR*, 2021, pp. 1–25.
- [20] S. Bai, F. Zhang, P.H.S. Torr, Hypergraph convolution and hypergraph attention, *Pattern Recognit.* 110 (2021) 107637.
- [21] H.K. Kim, J.K. Kim, Q.Y. Chen, A product network analysis for extending the market basket analysis, *Expert Syst. Appl.* 39 (8) (2012) 7403–7410.
- [22] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, LightGCN: Simplifying and powering graph convolution network for recommendation, in: *Proceedings of the SIGIR*, 2020, pp. 639–648.
- [23] B. Fu, W. Zhang, G. Hu, X. Dai, S. Huang, J. Chen, Dual side deep context-aware modulation for social recommendation, in: *Proceedings of the WWW*, 2021, pp. 2524–2534.
- [24] Z. Hu, Y. Dong, K. Wang, Y. Sun, Heterogeneous graph transformer, in: *Proceedings of the WWW*, 2020, pp. 2704–2710.
- [25] L. Yu, Y. Leng, D. Zhang, S. He, Collaborative group embedding and decision aggregation based on attentive influence of individual members: A group recommendation perspective, *Decis. Support Syst.* 165 (2023) 113894.
- [26] Z. Lin, J. Pan, S. Zhang, X. Wang, X. Xiao, S. Huang, L. Xiao, J. Jiang, Understanding the ranking loss for recommendation with sparse user feedback, in: *Proceedings of the KDD*, 2024, pp. 5409–5418.
- [27] Y. Lei, W. Li, Z. Lu, M. Zhao, Alternating pointwise-pairwise learning for personalized item ranking, in: *Proceedings of the CIKM*, 2017, pp. 2155–2158.
- [28] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: A survey, *Decis. Support Syst.* 74 (2015) 12–32.
- [29] Z. Xiao, H. Tong, Federated contrastive learning with feature-based distillation for human activity recognition, *IEEE Trans. Neural Netw. Learn. Syst.* (2025) 1–14.
- [30] G. Gao, H. Liu, K. Zhao, Live streaming recommendations based on dynamic representation learning, *Decis. Support Syst.* 169 (2023) 113957.
- [31] M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, in: *Proceedings of the RecSys*, 2010, pp. 135–142.

- [32] T. Lin, C. Gao, Y. Li, Recommender systems with characterized social regularization, in: Proceedings of the CIKM, 2018, pp. 1767–1770.
- [33] C. Zhang, L. Yu, Y. Wang, C. Shah, X. Zhang, Collaborative user network embedding for social recommender systems, in: Proceedings of the SDM, 2017, pp. 381–389.
- [34] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, M. Wang, A neural influence diffusion model for social recommendation, in: Proceedings of the SIGIR, 2019, pp. 235–244.
- [35] J. Liao, W. Zhou, F. Luo, J. Wen, M. Gao, X. Li, J. Zeng, Socialgn: Light graph convolution network for social recommendation, *Inf. Sci.* 589 (2022) 595–607.
- [36] W. Fan, Y. Ma, Q. Li, Y. He, Y.E. Zhao, J. Tang, D. Yin, Graph neural networks for social recommendation, in: Proceedings of the WWW, 2019, pp. 417–426.
- [37] Q. Wu, H. Zhang, X. Gao, P. He, P. Weng, H. Gao, G. Chen, Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems, in: Proceedings of the WWW, 2019, pp. 2091–2102.
- [38] X. Xiao, J. Wen, W. Zhou, F. Luo, M. Gao, J. Zeng, Multi-interaction fusion collaborative filtering for social recommendation, *Expert Syst. Appl.* 205 (2022) 117610.
- [39] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, X. Sun, Measuring and relieving the over-smoothing problem for graph neural networks from the topological view, in: Proceedings of the AAAI, 2020, pp. 3438–3445.
- [40] A. Antelmi, G. Cordasco, M. Polato, V. Scarano, C. Spagnuolo, D. Yang, A survey on hypergraph representation learning, *ACM Comput. Surv.* 56 (1) (2024) 24:1–24:38.
- [41] J. Han, Y. Tang, Q. Tao, Y. Xia, L. Zhang, Dual homogeneity hypergraph motifs with cross-view contrastive learning for multiple social recommendations, *ACM Trans. Knowl. Discov. Data* 18 (6) (2024) 158:1–158:24.
- [42] M.J. Salganik, P.S. Dodds, D.J. Watts, Experimental study of inequality and unpredictability in an artificial cultural market, *Science* 311 (5762) (2006) 854–856.
- [43] P. Bonhard, M.A. Sasse, 'Knowing me, knowing you'—Using profiles and social networking to improve recommender systems, *BT Technol. J.* 24 (3) (2006) 84–98.
- [44] M. Ruef, H.E. Aldrich, N.M. Carter, The structure of founding teams: Homophily, strong ties, and isolation among US entrepreneurs, *Am. Sociol. Rev.* 68 (2) (2003) 195–222.
- [45] A. Styhre, The role of social capital in knowledge sharing: the case of a specialist rock construction company, *Constr. Manag. Econ.* 26 (9) (2008) 941–951.
- [46] D. Castiglione, J.W. Van Deth, G. Wolleb, *The Handbook of Social Capital*, Oxford University Press, 2008.
- [47] C. Chiu, M. Hsu, E.T.G. Wang, Understanding knowledge sharing in virtual communities: An integration of social capital and social cognitive theories, *Decis. Support Syst.* 42 (3) (2006) 1872–1888.
- [48] T. Yu, J. Guo, W. Li, M. Lu, A mixed heterogeneous factorization model for non-overlapping cross-domain recommendation, *Decis. Support Syst.* 151 (2021) 113625.
- [49] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: Proceedings of the ICLR, 2013, pp. 1–12.
- [50] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, J. Tang, Session-based social recommendation via dynamic graph attention networks, in: Proceedings of the WSDM, 2019, pp. 555–563.
- [51] S. Ge, Y. Chen, Confidence-aware multimodal learning for trustworthy fake news detection, *INFORMS J. Comput.* (2025) 1–24.
- [52] Y. Sun, Y. Fang, K.H. Lim, D.W. Straub, User satisfaction with information technology service delivery: A social capital perspective, *Inf. Syst. Res.* 23 (4) (2012) 1195–1211.
- [53] K. Lin, H. Lu, Intention to continue using facebook fan pages from the perspective of social capital theory, *Cyberpsychology Behav. Soc. Netw.* 14 (10) (2011) 565–570.
- [54] C. Chang, M. Hsu, Understanding the determinants of users' subjective well-being in social networking sites: an integration of social capital theory and social presence theory, *Behav. Inf. Technol.* 35 (9) (2016) 720–729.
- [55] Z. Xiao, H. Xing, R. Qu, H. Li, X. Cheng, L. Xu, L. Feng, Q. Wan, Heterogeneous mutual knowledge distillation for wearable human activity recognition, *IEEE Trans. Neural Netw. Learn. Syst.* (2025) 1–15.
- [56] T. Raeder, N.V. Chawla, Modeling a store's product space as a social network, in: Proceedings of the ASONAM, 2009, pp. 164–169.
- [57] Y. Xu, L. Zhu, J. Li, F. Li, H.T. Shen, Temporal social graph network hashing for efficient recommendation, *IEEE Trans. Knowl. Data Eng.* 36 (7) (2024) 3541–3555.
- [58] C. Chen, M. Zhang, Y. Liu, S. Ma, Social attentional memory network: Modeling aspect- and friend-level differences in recommendation, in: Proceedings of the WSDM, 2019, pp. 177–185.
- [59] C. Chen, M. Zhang, C. Wang, W. Ma, M. Li, Y. Liu, S. Ma, An efficient adaptive transfer neural network for social-aware recommendation, in: Proceedings of the SIGIR, 2019, pp. 225–234.
- [60] X. Wang, X. He, M. Wang, F. Feng, T. Chua, Neural graph collaborative filtering, in: Proceedings of the SIGIR, 2019, pp. 165–174.
- [61] L. Chen, L. Wu, R. Hong, K. Zhang, M. Wang, Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach, in: Proceedings of the AAAI, 2020, pp. 27–34.
- [62] X. Wang, X. He, Y. Cao, M. Liu, T. Chua, KGAT: knowledge graph attention network for recommendation, in: Proceedings of the KDD, 2019, pp. 950–958.
- [63] Y. Wang, S. Tang, Y. Lei, W. Song, S. Wang, M. Zhang, Disenhan: Disentangled heterogeneous graph attention network for recommendation, in: Proceedings of the CIKM, 2020, pp. 1605–1614.
- [64] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, P.S. Yu, Heterogeneous graph attention network, in: Proceedings of the WWW, 2019, pp. 2022–2032.
- [65] C. Shi, B. Hu, W.X. Zhao, P.S. Yu, Heterogeneous information network embedding for recommendation, *IEEE Trans. Knowl. Data Eng.* 31 (2) (2019) 357–370.
- [66] J. Yu, H. Yin, J. Li, Q. Wang, N.Q.V. Hung, X. Zhang, Self-supervised multi-channel hypergraph convolutional network for social recommendation, in: Proceedings of the WWW, 2021, pp. 413–424.
- [67] T. Wang, L. Xia, C. Huang, Denoised self-augmented learning for social recommendation, in: Proceedings of the IJCAI, 2023, pp. 2324–2331.
- [68] Z. Li, L. Xia, C. Huang, RecDiff: Diffusion model for social recommendation, in: Proceedings of the CIKM, 2024, pp. 1346–1355.
- [69] C. Liu, J. Zhang, S. Wang, W. Fan, Q. Li, Score-based generative diffusion models for social recommendations, 2024, *CoRR*, arXiv:2412.15579.
- [70] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: Proceedings of the ICLR, 2019, pp. 1–19.
- [71] K. Yuan, G. Liu, J. Wu, H. Xiong, Semantic and structural view fusion modeling for social recommendation, *IEEE Trans. Knowl. Data Eng.* 35 (11) (2023) 11872–11884.
- [72] G. Oestreicher-Singer, B. Libai, L. Sivan, E. Carmi, O. Yassin, The network value of products, *J. Mark.* 77 (3) (2013) 1–14.
- [73] C.C. Chen, S. Shih, M. Lee, Who should you follow? Combining learning to rank with social influence for informative friend recommendation, *Decis. Support Syst.* 90 (2016) 33–45.
- [74] J. Zhang, K. Özpölat, G. Karamemis, D.G. Schniederjans, To disclose or not? The impact of prosocial behavior disclosure on the attainment of social capital on social networking sites, *Decis. Support Syst.* 192 (2025) 114437.
- [75] J.K. Yan, D.E. Leidner, H. Benbya, W. Zou, Social capital and knowledge contribution in online user communities: One-way or two-way relationship? *Decis. Support Syst.* 127 (2019).
- [76] X. Li, X. Guo, G. Chen, Integrating direct and indirect views for group recommendation: An inter- and intra-view contrastive learning method, *Decis. Support Syst.* 189 (2025) 114380.
- [77] J. Hsieh, Y. Fang, C.H. Liao, The power of choice: Examining how selection mechanisms shape decision-making in online community engagement, *Decis. Support Syst.* 182 (2024) 114250.

Weiyue Li received the B.S. degree from the School of Management, Hefei University of Technology, in 2021. He is currently pursuing the Ph.D. degree in Management Science and Engineering with the School of Management Science and Engineering, Dongbei University of Finance and Economics, China. His research interests include recommendation systems and social network analysis. His publications have appeared in IP&M, ESWA, among others.

Ming Gao received the BE, MS and Ph.D. degree in information technology and management from the School of Management Science and Engineering (SMSE), Dongbei University of Finance and Economics (DUFE), China, in 2002, 2004 and 2013. He is currently a professor and the head of department of big data management and application with SMSE, DUFE. He is also the deputy director of Key Laboratory of Big Data Management Optimization and Decision of Liaoning Province. His research interests include cloud computing, deep learning, and big data science and applications. He has published over 40 peer-reviewed journal and conference papers such as in *IEEE Trans*, *ACM TOIS*, *JASIST*, *JIL*, *AdvSci*, *BIOINF* and *AAAI*.

Bowei Chen received the Ph.D. in Computer Science from University College London. He is a Professor of Business Analytics and Artificial Intelligence at the Adam Smith Business School, University of Glasgow. He has broad research interests in the applications of probabilistic modeling and machine learning in business. His research has been published in venues such as *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Network Science and Engineering*, *ACM Transactions on Knowledge Discovery from Data*, *ACM Transactions on Intelligent Systems and Technology*, *SIGKDD*, *SIGIR*, *AAAI*, *European Journal of Operational Research* and *Tourism Management*.

Jingmin An received the Ph.D. degree in Computer Science from School of Information Science and Technology, Dalian Maritime university, in 2024. He is currently a Post-doctoral Researcher in management science and engineering with Dongbei University of Finance and Economics, China. His major research interests include recommender system, ubiquitous computing, and knowledge engineering. His publications have appeared in *ACM TOIS*, *Information Sciences*, *ESWA*, among others.

Yeming (Yale) Gong is a Professor of Management Science at EMLYON Business School. He holds a Ph.D. in Management Science from Rotterdam School of Management, Erasmus University, Netherlands. He published 145 SCI/SSCI articles in journals, including MIS Quarterly, Production and Operations Management, Transporta-

tion Science, various IEEE Transactions, IISE Transactions, and ACM Transactions. He received the “2010 Best Paper Award in Design and Manufacturing” from IISE, 2019 “Outstanding Paper Award” from ASCOM, and 2024 “Emerald Literati Award” (Outstanding Paper).