This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Dual Social View Enhanced Contrastive Learning for Social Recommendation

Shixiao Yang, Zhida Qin, Enjun Du, Pengzhan Zhou, *Member, IEEE*, and Tianyu Huang

***Abstract*—Social recommendation (SocialRS), which utilizes user social information to improve recommendation performance, has received increasing attention. Graph neural networks (GNNs) facilitate the integration of both user preference and social features in SocialRS. However, existing techniques face two challenges: 1) the inherent sparse supervision signals and noise issues in real-world social networks; 2) current social recommendation methods suffer from the neglect of user preference and social attribute heterogeneity, which hinders the extraction of preference-related information from social networks. Taking inspiration from social enhancement and contrastive learning methods, we propose a social recommendation model DSVC based on dual social view contrastive learning. Specifically, in response to the first challenge, our model derives the consistency factors of users in different augmented social views, which are used to highlight noise-resistant users and jettison preference-independent social relationships in social views. To address the second challenge, we adopt probability vectors generated from consistency factors. These vectors guide the cross-view augmentation process of the interaction graph, which helps supplement social self-supervised signals and effectively avoid noise retained due to indiscriminate augmentation. The baseline model comparison experiment, ablation experiment, parameter adjustment experiment and robustness experiment conducted on three different real-world datasets consistently validated the effectiveness of our model in improving recommendation performance.**

***Index Terms*—Contrastive learning, graph neural network, social networks.**

I. Introduction

NOWADAYS, with the proliferation of online social platforms [1] and the increasing demand for exploring personalized user preferences, research on social recommender systems (SocialRS) has garnered significant attention. Formerly, numerous studies on social relationships have shown that people are more inclined to establish interpersonal relationships with those with similar preferences, a phenomenon known as *social homophily* [2]. Meanwhile, individuals who have formed social ties demonstrate a tendency to influence and assimilate with each other, leading to the development of more similar preferences, referred to as *social influence* [3]. Building upon the aforementioned two theories, SocialRS has been developed to leverage user-side implicit information effectively, alleviating the issues of data sparsity and the long-tail phenomenon in historical interaction data.

Early investigations [4], [5], [6], [7] have integrated social information as a social matrix into matrix factorization (MF) [8] to enhance collaborative filtering (CF) recommendations. Subsequently, recent efforts [9], [10], [11], [12], [13], [14] have applied graph neural networks (GNNs) to leverage the excellent learning ability for graph-structured data, achieving significant performance improvements. Correspondingly, characterized as graph structures capturing user trust relationships, social networks (SNs) have found expanding applications in graph-based recommendation frameworks. Despite the ability of GNNs to encode graph structures from user-item interaction graphs and SNs, there are unique challenges when dealing with SNs compared to traditional recommendations. The complex and trust-driven user relationships in SNs often cannot directly reflect user preference characteristics for the items. Furthermore, it is imperative to contemplate the integration of user representations that are heterogeneously encoded from social and interaction data. Therefore, these discrepancies demand methods that can overcome the constraints of traditional GNN-based approaches. Concerning this topic, the follow-up researches [15], [16], [17], [18], [19], [20] have made efforts to enhance the integration, scalability, and interpretability of GNN-based SocialRS.

The forefront SN-based research routes can be divided into three categories. Firstly, some studies (e.g., DESIGN [21] and DGNN [22]) ameliorate the encoding scheme of social information, enabling enhanced SN-based encoders to better integrate with the main recommendation task. Secondly, to compensate for the sparsity of social networks, some works (e.g., MHCN [23] and SEPT [24]) add auxiliary self-supervised tasks based on social networks to compensate for the deficiency of user-side supervision signals on user-item interaction graph. Thirdly,

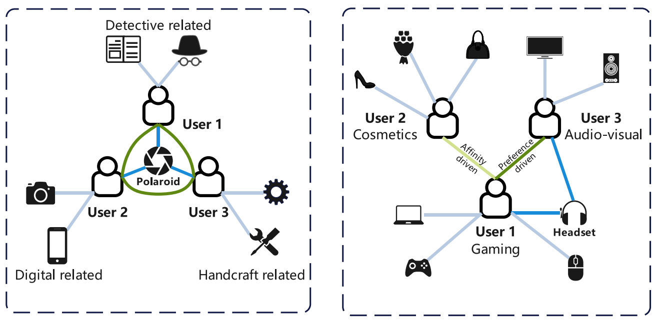
Received 29 January 2024; revised 30 September 2024 and 1 November 2024; accepted 8 November 2024. This work was supported by the National Nature Science Foundation of China under Grant 62102029 and Grant 62202072. *(Corresponding author: Zhida Qin.)*

Shixiao Yang, Zhida Qin, and Tianyu Huang are with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: ysx144_51@bit.edu.cn; zanderqin@bit.edu.cn; huangtianyu@bit.edu.cn).

Enjun Du is with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: enjundu.cs@gmail.com).

Pengzhan Zhou is with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: pzzhou@cqu.edu.cn).

Digital Object Identifier 10.1109/TCSS.2024.3496774



(a) (b)

Fig. 1. Preference-related scenarios based on: (a) stable; (b) preference-related social relationships.

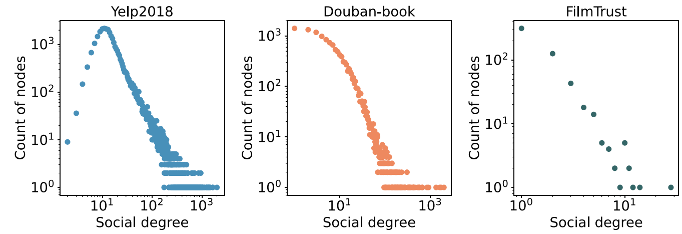


Fig. 2. The power-law distribution of users in real-world SNs.

when it comes to the issue of mass noise in social networks, partial works apply the structure-based [25] or feature-based [26] denoising method to affect the encoding process on SNs so that the SocialRS can adaptively pay less attention to invalid social information.

Despite their contribution to excavating social information, the aforementioned approaches still face two prominent challenges, including the heterogeneity of user preferences/social features and the inefficiency of extracting auxiliary signals from sparse social networks.

The first challenge inherently arises from the existence of irrelevant relations in social networks. Directly incorporating these relations into the user-item interaction graph will inject biased information into the recommendation task. Fig. 1 illustrates two recommendation scenarios based on stable and preference-related social relationships. In our study, we selected three users from the social network, forming a triangular relationship structure [Fig. 1(a)]. Notably, *User 1* prefers detective-themed items, *User 2* leans towards digital-related items, and *User 3* has a preference for items related to manual craftsmanship. Surprisingly, despite their diverse interests, the Polaroid camera, with its advantages in facilitating documentation, portability, and ease of disassembly, aligns with their shared preferences. In the second scenario [Fig. 1(b)], *User 1* maintains distinct social connections with *User 2* and *User 3*. However, their association is solely based on familial ties, leading to significant divergence in their item preferences. Conversely, *User 1* and *User 3* share a mutual interest in headsets, driven respectively by gaming and audio-visual pursuits. Based on the presented examples, we argue that encoding the social network structure superficially results in redundant social interactions, which hinder the accurate detection of interaction signals and consequently lead to a decline in recommendation performance. Furthermore, intimately social-associated users still exhibit restrictive overlap in preferred items due to significant individual differences.

Another challenge arises from the weak preference correlation faced by social networks, as well as sparsity similar to user-item interaction graphs. As shown in Fig. 2, the user and friend degree distribution in three real-world social network datasets follows a power-law pattern, with a small fraction of users with a lot of ”friends” and the majority of users in the long tail being not in vogue. This phenomenon results in limited effectiveness in incorporating social auxiliary information into recommendation tasks. Meanwhile, some studies [27], [28] have also pointed out that the sparsity of auxiliary social information has a negative impact on recommendation performance. Although SEPT [24] and similar approaches have attempted to address social information sparsity through auxiliary self-supervised learning (SSL) tasks on social networks, we argue that explicitly encoding social-supervised signals to enrich preference information is deemed highly limited in its efficacy. Specifically, these methods separate the tasks of social networks and user-item graph learning and only integrate them in the final stage of each training session, i.e., dual-tower-like structure, which lacks the direct impact of social networks on the generation of supervisory signals for recommendation tasks. Furthermore, the absence of early-stage augmented social-related guidance further hinders the complete utilization of advantageous social information.

For the first challenge, some studies have considered filtering irrelevant social relationships, such as MHCN [23] using hypergraphs and gating mechanisms to extract social paradigms that are conducive to recommendation, GDMSR [25] using preference similarity measurement to choose between dropout and retaining social relationships. Even with contributions, these methods are based solely on manual definition and similarity analysis to quantify user connections, ignoring the capture of fine-grained preference-social related stable information of social users, which reflects the consistency of user relationships in both interaction and social contexts; For the second challenge, other studies [24], [26] have taken into account the sparsity of SNs and used SSL methods to optimize users’ social representations. Although these approaches are involved in addressing the sparsity of SNs, they also face the impact of irrelevant social information, as the direct integration of social encoding and interaction encoding tasks cannot distinguish preference-related connections in SNs. Accordingly, we can adopt user interaction history to assist in the augmentation process of social data, thereby facilitating the extraction of reliable social supervision signals, and thereby reducing the impact of sparsity and irrelevant social connections.

To address the above-mentioned challenges, we propose a social recommender system based on dual social view enhanced contrastive learning for social recommendation (DSVC). Speaking separately, in response to the challenge of user feature disparity, we adopt a ternary closure augmentation strategy to filter and purify social information. Furthermore, to optimize the acquisition of robust users whose social-related information exhibits strong resistance to interference, we leverage contrastive learning (CL) among augmented social views to generate self-supervised signals. Then, we compute the consistency factor of the graph structure for individual users by utilizing diverse augmented user embeddings, which serve as a compressed representation of self-supervised information extracted from the aforementioned robust users on social networks. On the other hand, to address the challenge of insignificant auxiliary extraction, we propose a probability-guided strategy that selectively engages in the contrastive learning process of the user-item interaction graph, guided by consistency factors. This strategy serves as a bridge between the main task and auxiliary tasks, generating more distinct social-augmented signals from the early stages of the main task, thereby facilitating their integration. Ultimately, we use a series of Top-K ranking evaluation metrics to measure the recommendation performance of the model.

In summary, our contributions can be summarized as follows:

1) We propose a framework that leverages self-supervised signals generated through social networks to replace the conventional dual-tower-like structure. This framework not only efficiently harnesses social information but also significantly mitigates the impact of injecting social noise into collaborative filtering recommendations.
2) We propose a methodology to generate two augmented views of the original social network to alleviate noise and explore user features. Moreover, we introduce the consistency factor to measure users' resistance to noise interference.
3) We map the consistency factor to a probabilistic form for each user that guides the process of interaction graph augmentation. Thus, we utilize the generation process of auxiliary self-supervised signals to influence the main recommendation task.
4) We implement extensive experiments conducted on multiple real-world datasets to demonstrate the universality advantage of the proposed model. Comparative experiments with several other baseline models can demonstrate the unique advantages of the model. We also conduct robustness experiments to demonstrate the performance of each module in our model.

II. Related Work

A. Social Recommendation

The ascendancy of online social networks amplifies the importance of optimizing the utilization of user-side social relationship information. The presence of social homophily and social influence within social relationships significantly enhances the effectiveness of recommendations.

Notably, GNN has shown significant ability in capturing recursive social relationships between users, making it a popular method for modeling social relationships in recommender systems. Establishing a social relationship model requires addressing two key challenges: effectively describing the influence of different user neighbors, and seamlessly integrating interactive and social information.

In response to the first challenge, DiffNet [16] utilized an average pooling operation to ensure that each friend has the same influence on the target user. However, the influence of friends' preferences on users in the real world is inevitably imbalanced. Therefore, GraphRec [15] and DANSER [17] combined with graph attention networks achieved better performance improvement compared to DiffNet, further validating the hypothesis that different friends have different influences. In addition, due to the inherent noise of social information, ESRF [18] utilizes an automatic encoder mechanism to filter irrelevant social relationships and attempts to construct new neighbors to combat the noise of social information. Furthermore, SPEX [29] effectively combines the influence of social homophily and modeling it through multitasking learning and GNNs. EIISRS [30] proposes a social influence learning model to derive basic influence patterns in user relationships, overcoming sparsity while also modeling the influence of neighbors. BiasRec [31] utilizes the calculated bias score to mitigate the effect of non-preference factors, thereby addressing the inherent issues of data sparsity and preference bias in SocialRS. Our approach diverges from existing methodologies by considering the differences in users' sensitivity to noise and further characterizing the social stability of users under different social views through consistency factors. This not only distinguishes the influence of different users, but also provides supervision for subsequent user-item interaction data augmentation via the transformation of probability vectors.

For the second challenge, the mainstream method integrates user-item interaction information and social information by employing multiview joint training or hypergraphs. In recent years, SEPT [24] has adopted ternary self-supervised training to capture more supervised signals; MHCN [23] uses a hypergraph to improve complex social relationships and enhances social recommendations by utilizing higher order user relationships. To further alleviate the problems of overfitting and noise, DESIGN [21] combines knowledge distillation [32] as a novel cosupervision concept with social networks for the first time. GDMSR [25] introduces a preference-guided denoising framework that optimizes social graph denoising during the recommendation model learning process by modeling the confidence of social relationships. In contrast, our method incorporates the social stability of users as a guiding factor, directly influencing the augmentation process of the interaction graph, thereby facilitating the optimization of auxiliary contrastive learning tasks. DSL [26] preserves useful social relationships to enhance user-item interaction modeling and enables personalized cross-view knowledge transfer through adaptive semantic alignment in the embedding space. In contrast, our approach utilizes users' social embeddings solely for the probability-guided process, avoiding the residual social noise caused by incomplete denoising during joint optimization from interfering with recommendation tasks. In the latest work, ESGL [33] introduces a scale regularization module and joint SSL strategies to address the problems of vector scale distortion and over-smoothing. GBSR [34] learns denoised social structures through information bottleneck

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

methods, and identifies and reduces redundant social relationships. TMBCL [35] proposes a kind of time-aware graph neural network that incorporates temporal information and multibehavioral interactions into SocialRS. To sum up, rather than treating social encoding as a solitary task like the methods mentioned above, we transform the consistency of various augmented social information into guiding factors that serve as a bridge to direct the augmentation process of the user-item interaction view.

B. Contrastive Learning for Recommendation

Contrastive learning (CL) [36], [37], emerging as a prevalent SSL paradigm in recent years, maximizes mutual information [38] through the optimization of positive and negative sample pairs. As an upstream task, the CL method facilitates the acquisition of a discriminative representation space, wherein similar samples are mapped to proximate regions while retaining distinctiveness for dissimilar samples, effectively enhancing performance in downstream tasks. Recent investigations [39], [40], [41], [42] in contrastive learning have indicated that graph structures undergoing subtle perturbations exhibit comparable semantic properties. By comparing views derived from diverse perturbations, shared invariances are acquired through the utilization of structural perturbations as self-supervisory signals. Prominent variations of graph perturbations encompass uniform node dropout, edge dropout, and random walks, among other techniques.

According to the classification method in the work [43], views with different structures can be divided into same-scale contrast and cross-scale contrast. The strategy of same-scale contrast is currently widely used in recommendations and is usually further subdivided into local–local and global–global. For local level contrast, the dropout operation is usually used to form a local disturbance view. SGL [44] adopts three dropout methods to obtain augmented subgraphs. Subsequently, some work [45], [46], [47] have carried out a series of variant operations on SGL, such as changing the augmentation method or simplifying coding. HHGR [48] adopts the double-scale node dropout method and applies it to group recommendation scenarios. PCRec [49] uses the random walk strategy to sample the ego network to generate self-supervised signals. For global level contrast, as an architecture for generating global views, it is commonly used in sequence recommendations. CL4SRec [50] uses augmentation operators and generates negative sample sequences for input to the encoder to generate sequences and user representations. DHCN [51] introduces hypergraphs to generate views of different sessions and model positive and negative samples.

III. Problem Formulation

In this section, we provide formulaic definitions for two necessary data structures, i.e., social network and user-item interaction graph. Following that, we articulate a comprehensive problem formulation that underlies our social recommendation framework. To enhance clarity, Table I summarizes some key notations and definitions used in this article.

TABLE I
Summary of Key Notations and Definitions

Notation	Definition
$\mathcal{U},\ \mathcal{I}$	Set of users and items
m, n	Number of users and items
$\mathcal{G}_r,\ \mathcal{G}_s$	User-item interaction graph and social networks
$\mathcal{V}_u,\ \mathcal{V}_i$	Nodes of users and items
$\mathcal{E}$	Edges connecting users with interacted items in user-item bipartite graph
$\mathcal{E}_u$	Edges of social relationships between users in social networks
$\mathcal{R},\ \mathcal{S}$	User-item adjacency matrix and user–user adjacency matrix
$\mathbf{e}_u^0,\ \mathbf{e}_i^0$	Initial embeddings of users and items
Θ	Set of model parameters
ρ_{sn}	Hyperparameter for initial social view augmentation adjustment
$\widetilde{\mathcal{G}}_1,\ \widetilde{\mathcal{G}}_2$	Two randomly augmented social views
$\mathcal{A}_F,\ \mathcal{A}_S$	The ternary social view and the sharing view
$\mathcal{M}_u,\ u \in \mathcal{U}$	Set of other users in the view that are associated with a certain user
L	Number of propagation layers
$\phi(\cdot)$	Social view encoder
$s(\cdot)$	Cosine similarity function
$c_u^F,\ c_u^S$	Structural consistency factors generated from two augmented social views
p_τ	Hyperparameter for min–max normalization adjustment
ρ_a	Hyperparameter for the user-associated probability adjustment
$p_u^F,\ p_u^S$	Probability vectors generated from ternary social view and sharing view
$\widetilde{\mathcal{R}}_1,\ \widetilde{\mathcal{R}}_2$	Two augmented user-item interaction views
$\mathcal{N}_u,\ u \in \mathcal{U}$	Set of items that interact with a certain user
$\mathcal{N}_i,\ i \in \mathcal{I}$	Set of connected users for a certain item
$\mathbf{x}_u,\ \mathbf{x}_i$	Embeddings of users and items in final output
$\hat{r}_{ui}$	Final recommendation score for a certain user and a certain item
τ	Temperature hyperparameter for contrastive learning loss
β	Hyperparameter for contrastive learning loss weight adjustment

User-Item Interaction Graph. Assuming there are m users in the user set $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$, and n items forming the item set $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$. The user-item interaction graph is defined as $\mathcal{G}_r = \{\mathcal{V}_u, \mathcal{V}_i, \mathcal{E}\} : \mathcal{U} \times \mathcal{I}$, where $\mathcal{V}_u$ and $\mathcal{V}_i$ represent the user and item nodes, respectively, and $\mathcal{E}$ represents the edges connecting users and items that share an interactive relationship. Correspondingly, to facilitate storage and computation of $\mathcal{G}_r$, we define $\mathcal{R} \in \mathbb{R}^{m \times n}$ as user-item interaction matrix, where element $r_{u,i} = 1$ in the matrix represents that user $u \in \mathcal{U}$ has had interaction with item $i \in \mathcal{I}$, otherwise $r_{u,i} = 0$.

Social Network. From a dataset encompassing social networks defined as $\mathcal{G}_s = \{\mathcal{V}_u, \mathcal{E}_u\} : \mathcal{U} \times \mathcal{U}$, we can also extract social relationships among users who have item interactions to form the social matrix $\mathcal{S} \in \mathbb{R}^{m \times m}$, where element in the matrix $s_{i,j} = 1$ indicates that there is a social relationship between two users $u_i, u_j \in \mathcal{U}$. In contrast, a value of $s_{i,j} = 0$ signifies the absence of a social relationship between the two entities.

Problem Statement. We formally define the task as follows: *Input:* user-item interaction data $\mathcal{R}$ and social relationship data $\mathcal{S}$. *Output:* A learning function $\mathcal{F} = (u, i | \mathcal{R}, \mathcal{S}, \Theta)$ that used to predict the set of items that user $u (u \in \mathcal{U})$ would like to interact with, where Θ represents the model parameters.

IV. Methodology

The DSVC structure proposed in this section is shown in Fig. 3. The overall model can be divided into three parts:

1) *Calculation of Consistency Factors:* It introduces social networks to supplement user-side information and generates two augmented views using ternary closure augmentation to explore implicit social relationships. We further generate consistency factors between the user representations encoded from the social views.
2) *Social Relations Guided Augmentation:* It utilizes an innovative probability-guided node dropout approach,

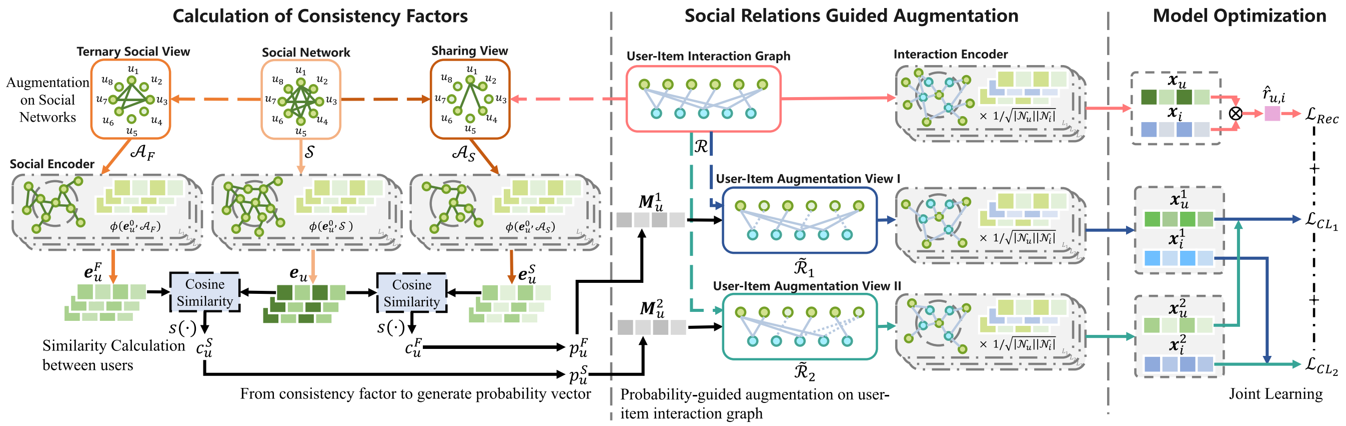


Fig. 3. Overall architecture of DSVC. The calculation of consistency factors part contributes to the augmented user representation consistency information, while the social relations guided augmentation part provides probability-guided interaction information. Subsequently, the model optimization part integrates the auxiliary self-supervised task and recommendation task and then optimizes the model parameters.

wherein the inclusion or exclusion of each user node is determined by the specific consistency factor, rather than the random augmentation strategy employed in traditional contrastive learning methods.

3) *Model Optimization:* It integrates the primary recommendation task with an auxiliary self-supervised task and iteratively optimizes the model parameters to attain enhanced recommendation performance.

A. Calculation of Consistency Factors

1) Social View Discrimination Process: In user social data, high-quality social relationships are significantly beneficial for recommendations, as there exists a strong preference influence among users who establish relationships. However, recent work [25] suggests that in real-world SNs, most social relationships are formed independently of user preferences, which suggests they are typically not beneficial for recommendation purposes. Integrating all social ties without discrimination is likely to inject noise and subsequently diminish recommendation performance. Inspired by prior works [44], [52] that focus on generating diverse views through self-discrimination for contrastive learning, we generate two augmented social views on the original social network by randomly dropout edges. The formula is expressed as follows:

$$\widetilde{\mathcal{G}}_1 = (\mathcal{V}_u, \mathbf{M}_1 \odot \mathcal{E}_u),\ \widetilde{\mathcal{G}}_2 = (\mathcal{V}_u, \mathbf{M}_2 \odot \mathcal{E}_u) \tag{1}$$

where $\widetilde{\mathcal{G}}_1, \widetilde{\mathcal{G}}_2$ respectively represent two augmented views; $\mathcal{V}_u$ represents the user node of the original social view $\mathcal{S}$; $\mathcal{E}_u$ is the edges between nodes, representing the social relationships between users; $\odot$ represents element-wise product; $\mathbf{M}_1, \mathbf{M}_2 \in \{0,1\}^{|\mathcal{E}_u|}$ is the masking vector for dropping edges, and its generation probability is controlled by hyperparameter ρ_{sn}.

2) Generation of Augmented Social Views: We use the ternary closure operation to construct two augmented views, including the ternary social view, which captures stable triangle social relationships among users, and the sharing view, which represents social connections between users who share interacted items. To elucidate the concept that refers to the maintenance of the integrity of reliable social relationships while mitigating noise, we offer the following two explanations. First, from the perspective of social influence, the ternary social view is interpreted as reflecting users' interest in expanding their social circles, while the sharing view describes users' interest in sharing preferred items with friends. Second, from the perspective of social homophily, if two individuals share common friends or interests in the same item, they are more likely to establish connections. Hence, these justify the use of the ternary closure operation to augment crucial ties. To further explain the form of ternary closures, we give the following examples:

From the intuitive example in Fig. 3, we can observe that taking the users u_1, u_3, u_6 that were not dropped during the augmentation process as an example. If they are "friends" with each other, the edges between these three users will be preserved in the ternary social view. Similarly, if users u_1, u_6 have interacted with the same item in the user-item interaction graph, the edge between these users will be preserved in the sharing view.

We can use matrix multiplication to extract the above two types of views. Let $\mathcal{A}_F \in \mathbb{R}^{m\times m}$ and $\mathcal{A}_S \in \mathbb{R}^{m\times m}$ represent the user adjacency matrix involved in these two types of relationships. They can be calculated by

$$\mathcal{A}_F = (\widetilde{\mathcal{G}}_1 \cdot \widetilde{\mathcal{G}}_1) \odot \widetilde{\mathcal{G}}_1,\ \mathcal{A}_S = (\mathcal{R} \cdot \mathcal{R}^\top) \odot \widetilde{\mathcal{G}}_2 \tag{2}$$

where the purpose of using matrix multiplication operations $\widetilde{\mathcal{G}} \cdot \widetilde{\mathcal{G}}$ and $\mathcal{R} \cdot \mathcal{R}^\top$ is to accumulate the paths that connect two users by sharing friends (items), and then make these paths into a triangle through Hadamard product $\odot \widetilde{\mathcal{G}}$.

According to the above paradigm, the original social view is augmented to generate the ternary social view representing strong social relationships and the sharing view representing common preferences. To efficiently extract high-order user features from these three views, we use LightGCN as the social encoder to transform initial user embedding into a higher order and trainable form. The formula definition of a social encoder is as follows:

$$\mathbf{e}_u^{(l)} = \sum_{u \in \mathcal{M}_u} \frac{1}{|\mathcal{M}_u|} \mathbf{e}_u^{(l-1)} \tag{3}$$

where $\mathcal{M}_u$ represents the set of other users in the view that are associated with user u; $\mathbf{e}_u^{(l)}$ represents the user embedding representation of layer l, and the final user embedding representation $\mathbf{e}_u$ output by the encoder is obtained by weighted average calculation of each layer embedding. As illustrated in Fig. 3, the social encoder performs multilayer graph convolution operations with the initial user embedding e_u^0. It operates individually on the original view $\mathcal{S}$, the ternary social view $\mathcal{A}_F$, and the sharing view $\mathcal{A}_S$, resulting in $\mathbf{e}_u$, $\mathbf{e}_u^F$, and $\mathbf{e}_u^S$, respectively.

B. Social Relations Guided Augmentation

1) User Consistency Factor Generation Process: Inspired by recent research on graph structure consistency [53], [54], our DSVC aims to explore the consistency attributes of each user based on augmented views, which measure their resistance to noise effects. Specifically, we define c_u^F as the consistency factors between the user embeddings generated from the ternary social view and the original social view, while c_u^S represents the consistency factors between the user embeddings generated from the sharing view and the original social view. The consistency factors between representations encoded from different views are calculated as follows:

$$\begin{aligned} c_u^F &= s(\phi(\mathbf{e}_u^0,\ \mathcal{A}_F), \quad \phi(\mathbf{e}_u^0,\ \mathcal{S}))) \\ c_u^S &= s(\phi(\mathbf{e}_u^0,\ \mathcal{A}_S), \quad \phi(\mathbf{e}_u^0,\ \mathcal{S}))) \end{aligned} \tag{4}$$

where $\phi(\cdot)$ represents the social view encoder mentioned in (3); $\mathbf{e}_u^0$ represents the initial user embedding representation; $\mathcal{A}_F,\ \mathcal{A}_S$ represents the ternary social view and sharing view, and $\mathcal{S}$ represents the original social view; $s(\cdot)$ represents the cosine similarity function.

Based on the above definitions, it can be observed that when the consistency factor c_u associated with a user is large, their sensitivity to changes in topological information decreases, indicating a higher resistance to noise. Consequently, if user u is more susceptible to social network noise compared to another user u', it is more likely that their respective consistency factors will follow the relationship $c_u < c_{u'}$. Thus, the consistency factor represents the social stability of users under different social views. This not only integrates the robust features of different views to enhance model adaptability in sparse social networks, but also provides supervision for subsequent user-item interaction data augmentation through the transformation of probability vectors, indirectly alleviating its inherent sparsity.

2) Probability-Guided Augmentation: Recently, certain approaches [44] have been investigated for augmenting user-item interaction views using random dropout. However purely random augmentation is not conducive to preserving crucial structural information, and residual noise continues to mislead collaborative information. Users with higher consistency factors in social networks have stronger antiinterference ability to noise. Besides, they also have more potential contributions to modeling and characterizing the true interests of other users. Therefore, to capture important interaction structure information, we propose the following series of usability expansion processing for the consistency factor to generate a probability vector, which can better guide the dropout process on the interaction graph. Taking sharing view $\mathcal{A}_S$ as an example, its corresponding probability vector p_u^S represents the estimated probability of dropping user nodes u and all their edges on interaction graph $\mathcal{R}$. The formula is calculated as follows:

$$\begin{aligned} w_u^S &= \exp(c_u^S) \\ {p'}_u^S &= \max\left(\frac{w_u^S - w^{min}}{w^{max} - w^{min}}, p_\tau\right) \\ p_u^S &= \rho_a \cdot \mu_{p'} \cdot {p'}_u^S \end{aligned} \tag{5}$$

where w_u^S in the first sub-equation denotes the impact level exerted by alterations in social relationships on user u, which is directly proportional to the associated consistency factor c_u^S through the exponentiation operation; Subsequently, to determine user dropout probability, we use min–max normalization in the second sub-equation according to their corresponding impact level, preventing users from mistakenly discarded due to minor disturbances. Specifically, we introduce a truncation probability p_τ to mitigate the effects of low normalized values, thereby generating the intermediate variable ${p'}_u^S$, which reflects the proportional relationship between the normalized impact level and dropout probability. Finally, to generate the dropout probability p_u^S in the third subequation, we use the mean value $\mu_{p'}$ of the intermediate variable to counteract extreme values, and employ hyperparameter ρ_a to enable manual intervention. For the ternary social view, the probability vector p_u^F is obtained similarly.

In contrast to previous dropout approaches that treat nodes or edges equally, our method increases the likelihood of dropping out social users who are easily disturbed. This minimizes the inclusion of noisy relationships generated by these users, which can be unhelpful or even disruptive to recommendation performance. The probability vectors guide the generation of user-item augmented views, ensuring a greater proportion of stable social users whose preferences are more positively influenced by social interactions. This approach achieves stable social relationship mining and enhances the effectiveness of probability-guided interaction augmentation.

When the probability vectors are p_u^F and p_u^S, two masking vectors $\mathbf{M}_u^1,\ \mathbf{M}_u^2\ \in \{0,1\}^{|\mathcal{V}_u|}$ are further generated based on the Bernoulli distribution. The augmentation process of applying it to user-item interaction diagram $\mathcal{R} = \{\mathcal{V}_u, \mathcal{V}_i, \mathcal{E}\}$ is as follows:

$$\widetilde{\mathcal{R}}_1 = (\mathbf{M}_u^1 \odot \mathcal{V},\ \mathcal{E}), \qquad \widetilde{\mathcal{R}}_2 = (\mathbf{M}_u^2 \odot \mathcal{V},\ \mathcal{E}). \tag{6}$$

User nodes in $\mathcal{R}$ dropout connected interaction edges based on its corresponding probability, generating two augmented user-item interaction views $\widetilde{\mathcal{R}}_1$ and $\widetilde{\mathcal{R}}_2$, respectively defined as user-item augmented interaction view I and user-item augmented interaction view II.

C. Model Optimization

In the model loss optimization section, we design a joint embedding space shared by the main recommendation task and the auxiliary self-supervised task, which further couples the BPR loss and InfoNCE loss framework.

Based on the original and two augmented interaction views, we employ an interaction encoder based on LightGCN to generate representations of users and items. Specifically, we utilize a higher order message propagation mechanism to encode the collaborative information from user-item interactions. Assuming a pair of initial inputs are $(\mathbf{e}_u^0, \mathbf{e}_i^0)$, the formula for the interaction encoder is expressed as follows:

$$\mathbf{e}_u^{(l+1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} \mathbf{e}_i^{(l)}, \ \mathbf{e}_i^{(l+1)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i||\mathcal{N}_u|}} \mathbf{e}_u^{(l)} \tag{7}$$

where $\mathbf{e}_u^{(l)}$ and $\mathbf{e}_i^{(l)}$ represent the encoded representations of user u and item i on the lth propagation layer of the L layer graph. $\mathcal{N}_u$ and $\mathcal{N}_i$ represent the set of items that interact with user u and the set of connected users for item i, respectively. Finally, each layer of embedding is combined to form the final user and item embeddings $\mathbf{x}_u$ and $\mathbf{x}_i$.

In this approach, we feed the initial pairs of user and item embeddings $(\mathbf{e}_u^0, \mathbf{e}_i^0)$ into the interaction encoder. They independently undergo convolution operations with the original interaction graph, augmented interaction view I, and augmented interaction view II. After applying weighted averaging to the outputs at each layer, the final pair representations of the user-item embeddings are denoted as $(\mathbf{x}_u, \mathbf{x}_i)$, $(\mathbf{x}_u^1, \mathbf{x}_i^1)$, and $(\mathbf{x}_u^2, \mathbf{x}_i^2)$.

Then, we perform dot product prediction on the user and item representation $\mathbf{x}_u, \ \mathbf{x}_i$ generated based on the original interaction view encoding, and compute the final recommendation score $\hat{r}_{u,i}$ as follows:

$$\hat{r}_{u,i} = \mathbf{x}_u^\top \cdot \mathbf{x}_i. \tag{8}$$

Moreover, the calculation method for BPR loss $\mathcal{L}_{\text{rec}}$ maintained by the main recommendation task is as follows:

$$\mathcal{L}_{\text{rec}} \ = \sum_{u \in U} \sum_{i \in N_u} \sum_{i' \notin N_u} - \log \ \sigma(\hat{r}_{u,i} - \hat{r}_{u,i'}) \tag{9}$$

where $\sigma(\cdot)$ represents the sigmoid activation function; $i' \notin N_u$ represents any item that u has not interacted with, which is obtained through random sampling; $\hat{r}_{u,i}$ represents the predicted score for item $i \in \mathcal{N}_u$; $\hat{r}_{u,i'}$ is the rating of the model on any item that the user u has not interacted with.

In addition, the model performs contrastive learning on user and item representation $(\mathbf{x}_u^1, \mathbf{x}_i^1)$ and $(\mathbf{x}_u^2, \mathbf{x}_i^2)$ generated based on two augmented interaction view codes. Specifically, positive and negative samples are generated from two embeddings of user u or item i generated by two augmented views, and the MI of samples is maximized by minimizing the InfoNCE loss function, to achieve the effect of making positive samples closer and negative samples more distant. The InfoNCE losses $\mathcal{L}_{CL_1}$ and $\mathcal{L}_{CL_2}$ proposed for contrastive learning to optimize the representation of users and items, are computed separately as follows:

$$\begin{aligned}\mathcal{L}_{CL_1} &= \sum_{u \in \mathcal{U}} - \log \frac{\exp(s(\mathbf{x}_u^1, \ \mathbf{x}_u^2)/\tau)}{\sum_{u' \in \mathcal{U}, u' \neq u} \exp(s(\mathbf{x}_u^1, \ \mathbf{x}_{u'}^2)/\tau)} \\ \mathcal{L}_{CL_2} &= \sum_{i \in \mathcal{N}_u} - \log \frac{\exp(s(\mathbf{x}_i^1, \ \mathbf{x}_i^2)/\tau)}{\sum_{i' \in \mathcal{I}, i' \neq i} \exp(s(\mathbf{x}_i^1, \ \mathbf{x}_{i'}^2)/\tau)}\end{aligned} \tag{10}$$

where u' is a negative sample of user u, representing any other user except u in the training set; i' is a negative sample of item i, representing any item in the training set that the user u has not interacted with; τ is the temperature parameter; $s(\cdot)$ is the cosine function that used to characterize the similarity between the sampled pairs.

According to the above two loss functions, the comprehensive optimization loss of this model is

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \beta(\mathcal{L}_{CL_1} + \mathcal{L}_{CL_2}) + \lambda\|\Theta\|_2^2 \tag{11}$$

where the hyperparameter β is used to adjust the proportion of InfoNCE loss function $\mathcal{L}_{CL_1}, \mathcal{L}_{CL_2}$ from two augmented interaction views respectively in the whole loss function; Θ is the parameter set of our model, while L_2 regularization parameter λ can adjust its weight value to prevent overfitting. The overall algorithm flow of DSVC is shown in Algorithm 1.

Algorithm 1: The Algorithmic Procedure of DSVC

Input: User-Item Interaction Graph $\mathcal{G}_r$, Social Networks $\mathcal{G}_s$, Randomly generated node embedding $\mathbf{E}$

Output: Recommendation lists

```
Social view discrimination with Eq. (1);
Augmented social views A_F, A_S generation with Eq. (2);
for each iteration do
    Consistency factor generation c_u with Eq. (3)–(4);
    Probability vectors generation p_u and corresponding mask vectors M_u with Eq. (5);
    Probability guided user-item graph augmentation R~ with Eq. (6);
    for each batch do
        User and item embedding e_u, e_i generation with Eq. (7);
        Model prediction r^_{u,i} with Eq. (8);
        Joint optimization with Eq. (9)–(11);
    end
end
```

D. Model Analysis of DSVC

1) Analysis of Consistency Factor: In DSVC, we utilize the user's consistency factor to guide the process of user-item interaction graph augmentation. In this section, we conduct an in-depth exploration of the potential advantages of consistency factors in SSL tasks.

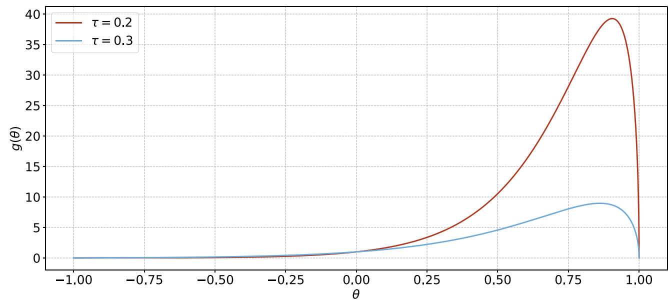


Fig. 4. Graphical trend of gradient function $g(\theta)$ under $\tau = 0.2$ and $\tau = 0.3$.

Assuming we use the consistency factor c_u to measure the consistency of user u. So we can define users with high consistency as

$$u_c = \{u | c(u) \gg \eta \geq \|c(u)\|\} \tag{12}$$

where η is the fixed threshold with user average consistency $\|c(u)\|$ as lower bound. For users with high consistency, it has a more stable feature representation.

For binary classification tasks, positive samples are defined as U_P, negative samples as U_N, and hard negative samples as U_H. We commonly consider negative samples exhibiting cosine similarities with positive samples within the range of $(\theta_0, 1]$ as hard negative samples, where $|\theta_0| < \epsilon$ is the lower bound for separating. From a combined perspective, users with high consistency in positive samples not only have easily distinguishable features, but also exhibit clear preferences and stable behavioral patterns. Accordingly, they have a lower likelihood of generating false hard negative samples compared to other users in positive samples. The above discussion can be expressed through probability formulas as follows:

$$\begin{aligned} &\mathcal{P}\{s(\mathbf{x}_u, \mathbf{x}_{u'}) > \theta_0 \mid c(u) \gg \eta,\ u' \in U_N \setminus U_H\} \\ &\quad < \mathcal{P}\{s(\mathbf{x}_u, \mathbf{x}_{u'}) > \theta_0 \mid c(u) \leq \eta,\ u' \in U_N \setminus U_H\} \\ &\mathcal{P}\{s(\mathbf{x}_{u_c}, \mathbf{x}_{u'_f}) > \theta_0\} < \mathcal{P}\{s(\mathbf{x}_u, \mathbf{x}_{u'_f}) > \theta_0\} \end{aligned} \tag{13}$$

where $u'_f = \{u' \mid u' \in U_N \setminus U_H\}$ is false hard negative samples. To further explain the reason for its judgment, we regard $\mathbf{x}_{u'_f}$ as the soft negative sample representation that mistakenly falls in the area belonging to hard negative samples during the similarity calculation with the positive sample representation. We hypothesize that such misclassification could potentially lead to a substantial gradient bias in the model optimization process.

To identify the main causes of gradient bias, we conduct in-depth discussions on the structure of contrastive loss. Based on previous works [44], we assume that the estimated similarity between user node u and its negative sample u' is $\hat{s}(\mathbf{x}_u, \mathbf{x}_{u'})$, abbreviated here as θ, and the obtained contrast gradient $g(\theta)$ can be expressed as

$$g(\theta) = \sqrt{1 - \theta^2} \exp\left(\frac{\theta}{\tau}\right). \tag{14}$$

As shown in Fig. 4, we present the trend graph of the gradient function under $\tau = 0.2$ and $\tau = 0.3$. It is evident from visual observation that the gradient of the model experiences a significant increase with a high similarity θ (e.g., $0.25 \leq \theta \leq 0.9$). Specifically, the gradient descends most rapidly when encountering hard negative samples. Consequently, contrastive loss compensates for the weakness of the BPR loss in effectively mining hard negative samples. However, the presence of false hard negative samples leads to a sharp increase in gradients, which in turn results in biased learning performance of the model. Drawing from our prior discussion on consistency factors, we can opt for stable user representations that more readily distinguish features as positive samples. This approach can minimize the model's likelihood of sampling false hard negative samples, thereby reducing the gradient bias effect.

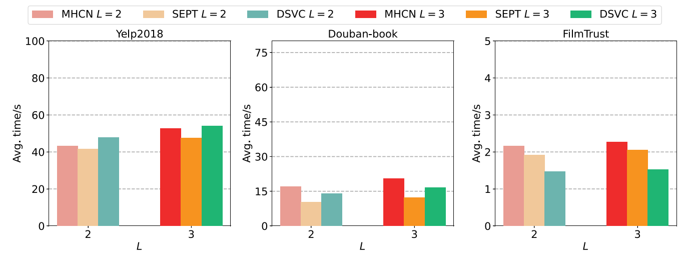


Fig. 5. Average running time comparison of DSVC, MHCN, and SEPT on 2 and 3 GNN layers.

2) Model Time Complexity Analysis: We analyze time complexity from the three key components of the DSVC framework. 1) For the social view generation module, we only need less than $4 \times \mathcal{O}(|\mathcal{S}| \times d)$ time complexity to generate the user embeddings through the social encoder, as both $\mathcal{A}_F$ and $\mathcal{A}_S$ are sparser than the original social view. Furthermore, due to the high sparsity shared by interaction and social data, the computational complexity of $\mathcal{A}_F$ and $\mathcal{A}_S$ based on sparse matrix multiplication is negligible. 2) For the social relations guided augmentation module, we only require $\mathcal{O}(2 \times |\mathcal{V}| \times d)$ time complexity to calculate the consistency factors. 3) For the model optimization module, the graph-based CF requires $\mathcal{O}(|\mathcal{E}| \times d)$ time complexity for modeling user-item interaction. The time complexity of calculating the InfoNCE loss is $\mathcal{O}(B \times (|\mathcal{U}| + |\mathcal{I}|) \times d)$, where B represents the number of unique users and items in the batch. Based on the above analysis, our DSVC achieves comparable time complexity when competing with state-of-the-art social recommendation models [23], [24]. To validate in practice, we train our method, MHCN, and SEPT with different GNN layer settings, and the average training time is shown in Fig. 5. Our method runs the fastest on FilmTrust and is the second fastest on Douban-book. Although it is slightly slower on Yelp2018 compared to the other models, we achieve significant improvements in recommendation performance within comparable time complexity.

V. Experiments

Conduct extensive experiments to verify the performance of our DSVC model by answering the following questions:

1) *RQ1:* How does our DSVC perform when competing with different types of recommendation methods?

TABLE II
STATISTICAL ANALYSIS OF THE DATASET USED IN THE EXPERIMENT

Datasets	Yelp2018	Douban-Book	FilmTrust
# User	45 919	13 025	1509
# Item	45 538	22 348	2072
# Interaction	1 183 610	598 420	35 497
# Relation	709 459	169 150	1853
U-I Density	5.66×10^{-4}	2.06×10^{-3}	1.14×10^{-2}
U-U Density	8.01×10^{-4}	1.04×10^{-3}	1.92×10^{-3}

2) *RQ2:* How do the different key modules in our DSVC framework contribute to overall performance?
3) *RQ3:* How do different hyperparameter settings (e.g., social network dropout rate ρ_{sn} and temperature constant τ, etc.) affect DSVC?
4) *RQ4:* How is the robustness of DSVC to the noise and sparsity disturbance compared to the existing models?
5) *RQ5:* How interpretable is DSVC in real situations?

A. Experiment Settings

1) Datasets: We conduct experiments on three public datasets collected from different real-life platforms: Yelp2018 for commercial venue recommendations, Douban-book for book recommendations, and FilmTrust for movie recommendations. Table I shows the statistical information of our experimental datasets, which have different interaction densities and social network characteristics.

2) Baselines: To explore the performance improvement of this model compared to other models using similar methods, we selected the following 12 recommendation models as the baselines for comparative experiments:

a) *MF* [8]*:* It mainly divides the interaction matrix into user and item matrices, and reconstructs the interaction matrix through relearning representation.
b) *TBPR* [55]*:* It introduces social networks and the concept of strong weak connections to mine more auxiliary social information.
c) *DiffNet* [16]*:* It proposes a deep influence propagation model to simulate the influence of higher order recursive social diffusion processes on users.
d) *LightGCN* [13]*:* It simplifies GCN and uses it in the collaborative filtering task of interaction graph, which has achieved significant performance improvement.
e) *ESRF* [18]*:* It utilizes a GCN-based adversarial network framework to capture higher order information and reduce the impact of noise, while also using ternary closures to augment relational data.
f) *Motifs-Res* [56]*:* While introducing self-supervised learning into social recommendation, this model also proposes a cross motif matching representation model combining attention mechanism and multichannel information.
g) *SEPT* [24]*:* This model constructs a social relationship awareness framework based on ternary training and self-supervised learning. It uses the user's social information to augment the social view and then builds three graph encoders on the augmented view.
h) *MHCN* [23]*:* It proposes a multichannel convolutional network based on hypergraph, and combines self-supervised learning to use high-order user relations to enhance social recommendation while compensating for aggregation losses. It is worth emphasizing that in the experiment section, we adopt the variant S^2-MHCN with a self-supervised learning module as our baseline.
i) *DESIGN* [21]*:* It proposes a social relationship graph augmentation network based on knowledge distillation, and uses both global and local views to train simultaneously.
j) *DSL* [26]*:* It proposes a self-enhanced learning framework that maintains crucial social connections and supports personalized cross-view knowledge exchange.
k) *DcRec* [52]*:* It approach derives disentangled user representations from both interaction and social spheres, utilizing contrastive learning to aid knowledge transfer between these representations, thereby improving social recommendations.
l) *LightGCL* [47]*:* It uses singular value decomposition for contrastive augmentation of the user-item graph, preserving semantic structures and enhancing robustness.

3) Evaluation Metrics: We perform a random partitioning of the user-item interaction datasets, allocating 80% of the data for the training set and reserving 20% for the testing set. The evaluation metrics used in the experiment are uniformly the hit rate HR@K, recall rate Recall@K, and normalized cumulative loss gain NDCG@K commonly used in Top-K ranking recommendation tasks, where we set K to 10 and 20.

4) Parameter Settings: Our DSVC is implemented using PyTorch with NVIDIA GeForce RTX 3090, while most baseline models used for comparative experiments are evaluated based on the SELFRec [57] framework. For the sake of fairness, the embedding representation dimension of all models is set at 64, and the learning rate is adjusted between $\{1e^{-4}, 1e^{-3}, 5e^{-3}, 1e^{-2}\}$ according to the different convergence rates of different models. The batch size is fixed at 2048.

In the fixed parameter section of the experiment, the dimension represented by each user and item node embedding is 64, the number of convolutional layers in both the social encoder and interactive encoder is $L = 3$, and the learning rate is $\alpha = 0.001$; In the variable hyperparameter section, the parameter ρ_{sn} used to control the social view edge drop process is adjusted within the range of $\{0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5\}$, the parameter ρ_a used to control the impact of the mean on probability in the probability generation vector generation process is adjusted within the range of $\{0.1,\ 0.12,\ 0.14,\ 0.16,\ 0.18,\ 0.2\}$, and the parameter τ used to control the proportion of hard negative sampling in the InfoNCE loss is adjusted within the range of $\{0.001,\ 0.005,\ 0.01,\ 0.05,\ 0.1,\ 0.5\}$. The adjustment range of parameter β used to control the proportion of InfoNCE loss in the total loss is $\{0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5\}$.

TABLE III
COMPARISON OF PERFORMANCE EVALUATION VALUES BETWEEN DSVC AND DIFFERENT BASELINE MODELS IN YELP2018, DOUBAN-BOOK, AND FILMTRUST DATASETS

Dataset	Metric	MF	TBPR	DiffNet	LightGCN	ESRF	Motifs-Res	SEPT	MHCN	DESIGN	DSL	DcRec	LightGCL	**DSVC**
Yelp2018	HR@10	0.02910	0.02240	0.03357	0.03626	0.03714	0.04012	0.03593	0.03326	0.39660	0.03313	0.03937	0.03924	**0.04239**
	Recall@10	0.03307	0.03125	0.03354	0.04116	0.04185	0.04815	0.04122	0.03649	0.04357	0.03120	0.04720	0.04821	**0.04855**
	NDCG@10	0.02750	0.03234	0.02051	0.03453	0.03526	0.04111	0.03465	0.03115	0.36600	0.02620	0.04000	0.04088	**0.04219**
	HR@20	0.04964	0.05953	0.05613	0.06036	0.06238	0.06246	0.06159	0.05894	0.06394	0.05158	0.06432	0.05847	**0.07118**
	Recall@20	0.05578	0.06286	0.06086	0.06730	0.06922	0.06945	0.06955	0.06581	0.07204	0.05170	0.07760	0.07909	**0.08028**
	NDCG@20	0.03592	0.04115	0.03834	0.04363	0.04395	0.04417	0.04476	0.04262	0.04614	0.03360	0.05070	0.05183	**0.05259**
Douban-book	HR@10	0.05778	0.07925	0.06646	0.07048	0.07338	0.07639	0.07310	0.07554	0.07332	0.06864	0.08067	0.07715	**0.08311**
	Recall@10	0.08728	0.08769	0.08836	0.07888	0.06546	0.07677	0.10272	0.10430	0.10402	0.05990	0.11400	0.10391	**0.11445**
	NDCG@10	0.09559	0.10720	0.10188	0.09887	0.10671	0.10274	0.12035	0.12370	0.09420	0.07240	0.13290	0.12778	**0.14051**
	HR@20	0.09415	0.10152	0.10234	0.11189	0.11435	0.11084	0.11545	0.11230	0.11625	0.09132	0.11130	0.11055	**0.12952**
	Recall@20	0.12902	0.13054	0.13110	0.14990	0.15301	0.15736	0.15470	0.15128	0.15873	0.09700	0.16100	0.14565	**0.17464**
	NDCG@20	0.10424	0.10460	0.10450	0.12741	0.12952	0.13395	0.13137	0.13257	0.13495	0.07950	0.13920	0.13296	**0.15287**
FilmTrust	HR@10	0.71403	0.71390	0.68001	0.72229	0.71955	0.71896	0.71875	0.72418	0.72300	0.72016	0.71926	0.63182	**0.72583**
	Recall@10	0.73226	0.59100	0.69427	0.73812	0.73521	0.51340	0.73219	0.73678	0.73534	0.71470	0.71988	0.71129	**0.74333**
	NDCG@10	0.57758	0.49920	0.55158	0.58478	0.57849	0.58670	0.57814	0.59677	0.59492	0.56520	0.56642	0.50598	**0.60012**
	HR@20	0.80778	0.82261	0.81983	0.82370	0.82418	0.82562	0.82724	0.82606	0.82574	0.81930	0.82684	0.77257	**0.83137**
	Recall@20	0.83779	0.84388	0.84628	0.85180	0.85236	0.85424	0.85432	0.85378	0.85254	0.82700	0.82426	0.70471	**0.86178**
	NDCG@20	0.60305	0.60853	0.61322	0.61507	0.61526	0.61691	0.61560	0.63294	0.61821	0.59830	0.59725	0.52832	**0.63827**

Note: The best results are displayed in bold, while the second-best results are highlighted below.

B. Comparison of Performance (RQ1)

In this part, we conduct an overall performance evaluation of DSVC and the mentioned baselines. Table III shows the experimental results of each model on three datasets, from which the following observations can be obtained:

1) Our DSVC consistently outperformed all baselines in three evaluation metrics, verifying the effectiveness of using augmented social networks as the guidance for comparative tasks. While the evaluation scenarios are diverse due to variations in interaction and social information characteristics, the consistently superior results highlight the broad applicability and versatility of DSVC.
2) SN-based baselines like DiffNet, ESRF, MHCN, and DESIGN directly incorporate social information into CF-based recommender systems. By comparing our DSVC with these models, we observe that DSVC effectively utilizes augmented social information as a guiding task for contrastive learning. This approach successfully avoids the inclusion of noisy and sparse social information in collaborative filtering, resulting in significant performance improvements.
3) By comparing the performance of Motifs-Res, SEPT, and MHCN within the SSL framework, we can observe that contrastive learning, which is widely adopted as a self-supervised learning paradigm, harnesses the benefits of augmenting supervisory signals and mitigating noise in recommender systems. Notably, concerning the baseline models SEPT and MHCN that incorporate both SSL and SNs, the success of our DSVC demonstrates that leveraging social networks for probability-guided augmentation on user-item interaction graphs has more effectiveness in crucial social information mining.

   By comparing the performance of DSL, DcRec, and LightGCL, we identify significant limitations in these models regarding the utilization of social information and the enhancement of robustness. DSL exhibits poor performance in the denoising process of social encoding, failing to effectively eliminate residual noise. DcRec introduces redundant social encoding, which constrains the identification of stable preferences. Meanwhile, LightGCL lacks effective integration of social supervision signals. In contrast, DSVC leverages probabilistic enhancement techniques to more effectively identify and utilize stable relationships within social networks, thereby providing more precise social guidance in contrastive learning.

C. Ablation Experiment (RQ2)

To explore whether each module in DSVC has a significant improvement in model performance, the ablation experiment section mainly conducts comparative experiments on three variants of the model:

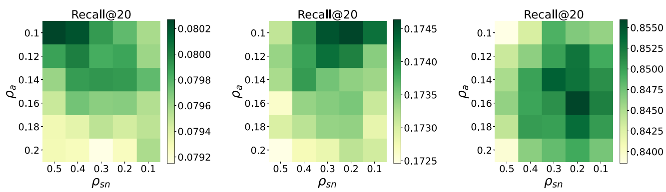
1) *w/o CL:* A variant of the original model that is discarded from the auxiliary SSL module. The model only works on the main recommendation task, which degenerates to LightGCN to a certain extent.
2) *w/o PB:* A variant of the original model that replaces the social probability-guided interaction view augmentation process with random augmentation. The model loses the ability to leverage augmented social self-supervised signals, resulting in the absence of the influence of the social network on the SSL process of the recommendation task.
3) *w/o DV:* A variant of the original model that only uses two randomly augmented social views instead of using ternary closure augmentation. The arbitrary augmentation approach largely retains numerous irrelevant and noisy social information, which in turn obstructs the subsequent robust assessment of users' susceptibility to noise.

Table IV shows the performance comparison of the original model and its three ablation variants under three evaluation metrics. From the ablation results, we can analyze and obtain the following observation results:

1) From the significant performance degradation observed in the variant "DSVC w/o CL" compared to the original model, we can observe that solely employing GCN to enhance the recommendation task is insufficient. The utilization of auxiliary self-supervised learning is necessary to better discover the latent information within

TABLE IV
ABLATION STUDY OF DSVC AND ITS VARIANTS ON REAL DATASETS

Model	Yelp2018			Douban-Book			FilmTrust		
Metrics@20	HR	Recall	NDCG	HR	Recall	NDCG	HR	Recall	NDCG
DSVC	**0.07118**	**0.08028**	**0.05259**	**0.12952**	**0.17464**	**0.15287**	**0.83137**	**0.86178**	**0.63827**
DSVC w/o CL	0.06114	0.06886	0.04500	0.11091	0.14770	0.12528	0.82193	0.84985	0.61584
DSVC w/o PB	0.07049	0.07946	0.05200	0.12677	0.17121	0.14862	0.82137	0.85415	0.63053
DSVC w/o DV	0.07042	0.07938	0.05182	0.12697	0.16970	0.14887	0.82370	0.85207	0.62650



(a) (b) (c)

Fig. 6. Impact of probability generation vector generation process related parameters ρ_{sn} and ρ_a. (a) Yelp2018. (b) Douban-book. (c) FilmTrust.

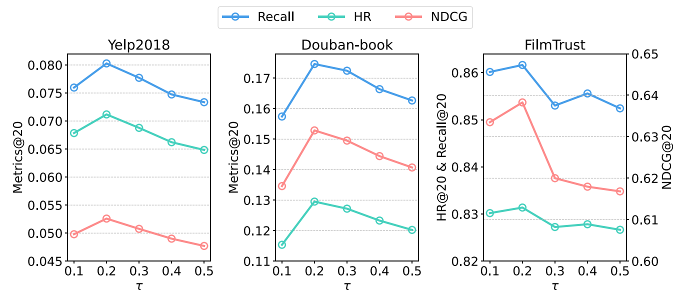


Fig. 7. Impact of temperature parameter τ.

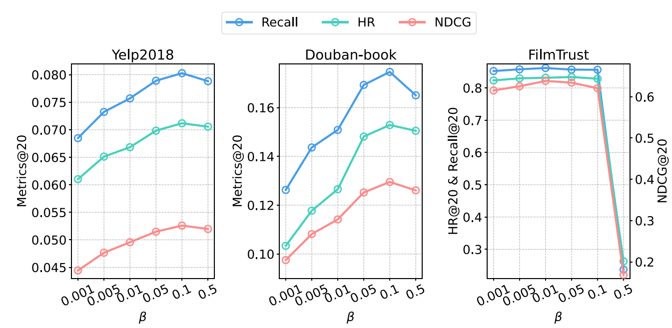


Fig. 8. Impact of weight β for contrastive loss.

the interaction graph, thereby exerting a more substantial influence on the recommendation task.

2) From the experimental results of the variant "DSVC w/o PB," it is evident that employing random augmentation on the interaction view may potentially disrupt robust collaborative signals and even retain biased information. As a result, this hinders the process of transferring critical self-supervised signals to the main recommendation task.
3) From the analysis of the experimental results of the variant "DSVC w/o DV," we can detect that using ternary closures to further augment the SN views after preliminary augmentation can filter out the individuals with strong resilience to disturbances in the remaining user nodes, thereby providing more accurate upstream information for the subsequent node consistency factor generation process.

D. Parameter Analysis (RQ3)

1) Impact of Probability Generation Vector Generation Process: Throughout the entire process of probability vector generation, the hyperparameters ρ_{sn} and ρ_a respectively play critical roles in the social view augmentation and the user-associated probability adjustment. Furthermore, we adjust ρ_{sn} within the range $\{0.1, 0.12, 0.14, 0.16, 0.18, 0.2\}$ and the ρ_a within the range $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ on all of three datasets. We analyze the distribution depicted in Fig. 6 separately. 1) For the Douban-book and FilmTrust, the value $\rho_{sn} = 0.2$ is optimal, while for Yelp2018, it is up to 0.5. This indicates that for views containing more noisy relationships, masking more users can achieve more ideal augmentation effects. 2) For Yelp and Douban-book, the value $\rho_a = 0.1$ is the best, while for FilmTrust, it turns to 0.16. This indicates that in scenarios with fewer socially active users, the ratio of robust users selected should also be appropriately reduced to prevent bias in the guidance results of downstream tasks.

2) Impact of Temperature Parameter: To investigate the impact of temperature parameter τ on the performance of contrastive learning, our parameter tuning experiments are conducted on Yelp2018, Douban-book, and FilmTrust datasets by setting ranges within $\tau = \{0.1, 0.2, 0.3, 0.4, 0.5\}$. From Fig. 7, it can be observed that the performance is best when $\tau = 0.2$ is present on all datasets. By observing and analyzing the results, it becomes evident that as the dataset size increases, the complexity and uncertainty of the data escalate. Consequently, valuable nodes are at risk of being discarded, while noisy nodes may persist. As a result, temperature parameter τ significantly influences the performance of the SSL task, particularly in scenarios involving larger data volumes.

3) Impact of Weight for Contrastive Loss: To explore the proportion β of InfoNCE loss in total loss to achieve optimal performance, our parameter tuning experiments are conducted on the Yelp2018, Douban-book, and FilmTrust datasets by setting ranges within $\beta = \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$. According to Fig. 8, we can observe that for Yelp and Douban-book, value $\beta = 0.1$ is ideal, whereas it comes to 0.01 for FilmTrust, and a severe underfitting problem was observed

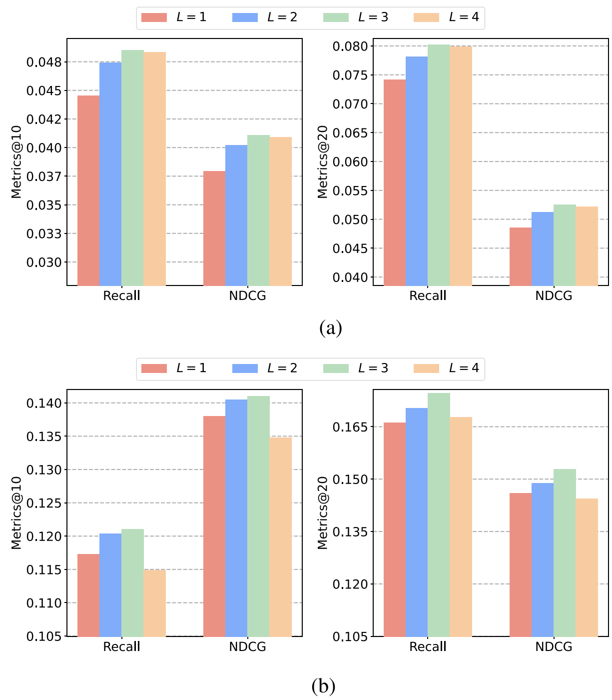


Fig. 9. Impact of GNN layers L for both social and interaction encoders. (a) Yelp2018 data. (b) Douban-book data.

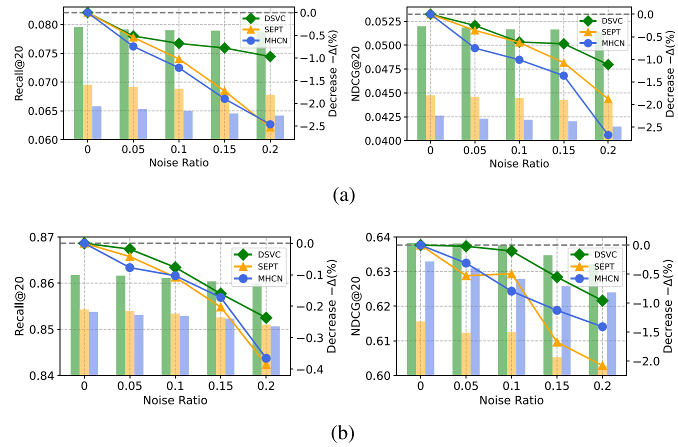


Fig. 10. Performance on Yelp2018 and FilmTrust datasets with noise perturbation in terms of Recall@20 and NDCG@20. (a) Yelp2018. (b) FilmTrust data.

when $\beta = 0.5$. Given that the primary objective of the model remains the collaborative filtering recommendation task based on GCN, it is crucial to strike a balance with controlling the lower proportion of self-supervised learning loss. Setting this proportion too high may lead to issues such as gradient vanishing or insufficient model performance due to the rapid descent of gradients.

4) Impact of GNN Layers: To explore the impact of GNN layers for social and interaction encoders on model performance, we conducted parameter adjustment experiments on the Yelp2018 and Douban-book datasets, with a range set within $L = \{1, 2, 3, 4\}$. According to Fig. 9, we can observe that when $L = 3$, all evaluation metrics reach their highest values, whereas the performance of $L = 2$ is significantly inferior. This result confirms the correctness and necessity of selecting $L = 3$ in our model design, indicating that appropriately adding one more propagation layer can enhance recommendation performance under our approach.

E. Model Robustness Study (RQ4)

In this section, we investigate the robustness of our DSVC approach to data noise and data sparsity by evaluating the model performance on manually damaged or segmented training data, in comparison with representative baseline methods.

1) Model Robustness Against Social Noise: To examine the robustness of DSVC in the presence of noise disturbances, we inject artificial noise edges of varying proportions (5%, 10%, 15%, and 20%) into the original social network through random injection. Subsequently, we evaluated the performance of our DSVC and two representative strong baselines, i.e., SEPT and MHCN, on distinct variants of Yelp2018 and FilmTrust datasets.

Fig. 10 illustrates that our DSVC method consistently exhibited the least performance degradation across all levels of noise. We attribute our observations to the following reasons: 1) Both MHCN and SEPT demonstrate a high sensitivity to noise across datasets of varying sizes. One plausible inference is that the SSL tasks in MHCN and SEPT transfer more fake social information from social networks to recommendation tasks. They both show weak supervision over fake social relations. 2) DSVC demonstrates a higher robustness to noise and achieves a less sensitive performance degradation, which can be ascribed to two underlying factors. First, our DSVC augments social views through two steps: random dropout and ternary closure, dualistically enhancing the model's ability to accurately identify robust users. Second, our SSL task essentially focuses on augmenting user-item interaction graphs instead of directly optimizing user representations generated from social views, and the probability-guided augmentation approach helps augmentation tasks avoid introducing noise from social networks. As a result, our SSL task exhibits low sensitivity to the perception of changes in social networks.

2) Model Robustness Against Social Sparsity: To examine the robustness of DSVC in different sparse social data scenarios, we divided the social network into subgraphs based on users in different social activity ranges (e.g., 0–10, 10–20, 20–40, 40–80, 80–200, 200–500, 500–2000) and ensured that there was few significant difference in the number of users between subgraphs.

Fig. 11(a) and 11(b) shows the performance changes of DSVC compared to SEPT and MHCN, while Fig. 11(c) shows the variance of performance σ^2 on both metrics. It can be observed that our DSVC method achieves minimal performance fluctuations. We attribute our observation results to the following reasons: 1) MHCN and SEPT exhibit sensitive performance changes to data with different sparsity levels, and have the highest performance reduction for user groups in the activity range of 40–80. We postulate that this phenomenon could be

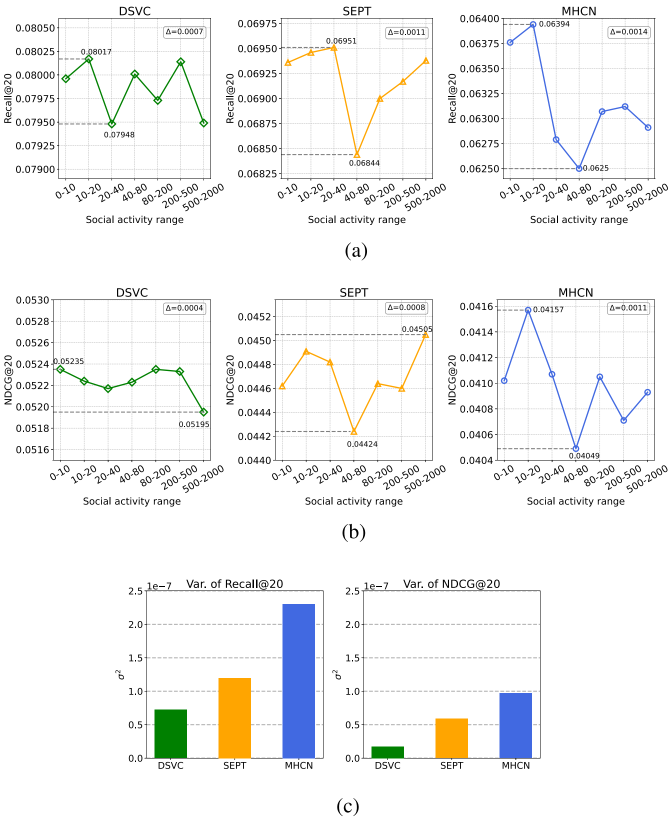


Fig. 11. Performance on Yelp2018 dataset with different sparse active user groups in terms of Recall@20 and NDCG@20. (a) Performance comparison on Recall@20. (b) Performance comparison on NDCG@20. (c) Comparison of overall variance of Recall@20 and NDCG@20.

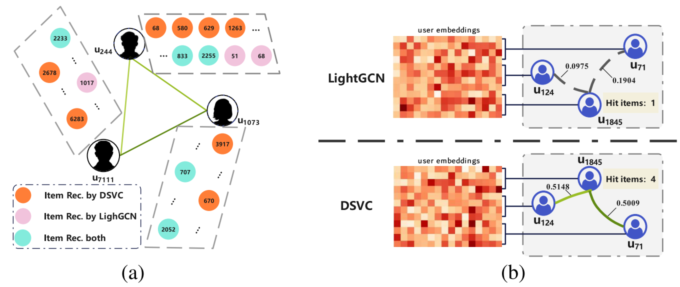


Fig. 12. Two typical examples of DSVC versus LightGCN for highlighting the effectiveness of social assistance. (a) Case study on recommendation results. (b) Case study on representation learning.

attributed to a decrease in the proportion of preference-related relationships among socially active users to the total relationship, i.e., an increase in the ratio of noise-related relationships, while the weak filtering mechanism of SEPT and MHCN limits their ability to identify internal noise information. 2) DSVC exhibits higher robustness to changes in social data with different sparsity and achieves less sensitive performance reduction, which can be attributed to our social view augmentation approach filtering out relationships irrelevant to recommendations to the maximum extent possible. Furthermore, Even as the severity of data sparsity has intensified, we can still capture stable users with high consistency, thereby reducing the bias impact on the contrastive task. Therefore, our social networks with different sparsity can exhibit relatively stable performance.

F. Case Study (RQ5)

To verify the effectiveness and interpretability of our DSVC in real-world scenarios, we extracted two typical examples from the Yelp2018 dataset. We compare these two examples with LightGCN in two aspects, i.e., recommendation results and representation learning, to illustrate the superiority of our method in integrating social networks into recommender systems.

1) Case Study on Recommendation Results: As shown in Fig. 12(a), one example extracted from the dataset is a set of stable ternary user relationships, represented by triple $(u_{244}, u_{7111}, u_{1073})$. In this ternary relationship, there are more than ten common interactive items between users in pairs, the stronger intensity of the color indicates the larger intersection of interacted items, which in turn implies higher preference similarity. The parallelogram-shaped box adjacent to each user displays the hit items, which coincide with the test set in the Top-20 results recommended by DSVC, LightGCN, and both of them.

For u_{244}, DSVC and LightGCN hit 6 and 4 items, respectively, and there is an intersection (e.g., i_{833} and i_{2255}); For u_{1073}, our LightGCN result is a true subset of the DSVC result and also includes a intersection (e.g., i_{707} and i_{2052}). From the recommendation results, it is evident that our DSVC significantly outperforms LightGCN in personalized recommendations due to its powerful ability to identify and extract stable relationships in social networks. Furthermore, items recommended by LightGCN are also effectively captured by our DSVC, showing the low interference ability of our auxiliary tasks for the main recommendation task performance.

2) Case Study on Representation Learning: Fig. 12(b) illustrates another example extracted from the dataset, involving user u_{1845} and two other users, u_{124} and u_{71}, who have established relationships with her. We separately visualize the final encoding representations of these three users generated from DSVC and LightGCN. Additionally, we calculate the cosine similarity between the representations of u_{1845} and u_{124}, as well as between u_{1845} and u_{71}.

In situations where social auxiliary signals are not available, models such as LightGCN determine the similarity between users indirectly only through their interaction history with the same items. Consequently, in the given example, users u_{124} and u_{71} are assigned a lower similarity, resulting in few hit items for user u_{1845}. In contrast, our DSVC leverages social relationships effectively, assigning higher similarity to user u_{1845} and two socially connected users with similar preferences. As a result, DSVC significantly outperforms LightGCN by the number of hit items.

VI. Conclusion

In this work, our proposed DSVC framework combines social networks and contrastive learning organically and attempts to alleviate data sparsity and noise issues. During the process of characterizing the social feature view, augmented social information was constructed to study auxiliary self-supervised signals. This work expands the application methods of social networks in the field of recommendation. Numerous experiments conducted on several real-world datasets have shown that DSVC has advantages over various advanced methods.

References

[1] H. L. J. Tang, X. Hu, "Social recommendation: A review," *Soc. Netw. Anal. Min.,* vol. 3, pp. 1113–1133, Dec. 2013.

[2] M. McPherson, L. Smith-Lovin, and J. M. Cook, "The enormous flock of homophily researchers: Assessing and promoting a research agenda," in *Personal Networks: Classic Readings and New Directions in Egocentric Analysis*, vol. 27, Cambridge, U.K.: Cambridge University Press, 2021, pp. 459–470.

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

[4] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst., (RecSys'10)*, 2010, pp. 135–142.

[5] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: Social recommendation using probabilistic matrix factorization," in *Proc. 17th ACM Conf. Inf. Knowl. Manage. (CIKM)*, 2008, pp. 931–940.

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

[7] G. Guo, J. Zhang, and N. Yorke-Smith, "TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 29, no. 1, 2015.

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

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

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

[11] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in Ind. Represent. Learn. Large Graphs Adv. Neur. Inf. Process. Syst. 30: Annu. Conf. Neur. Inf. Process. Syst. (NeurIPS), vol. 30, 2017.

[12] C. Feng, Z. Liu, S. Lin, and T. Q. Quek, "Attention-based graph convolutional network for recommendation system," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2019, pp. 7560–7564.

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

[14] F. Liu, Z. Cheng, L. Zhu, Z. Gao, and L. Nie, "Interest-aware message-passing GCN for recommendation," in *Proc. Web Conf.*, 2021, pp. 1296–1305.

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

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

[17] Q. Wu et al., "Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems," in *World Wide Web Conf. (WWW)*, 2019, pp. 2091–2102.

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

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

[20] C. Zhang, Y. Wang, L. Zhu, J. Song, and H. Yin, "Multi-graph heterogeneous interaction fusion for social recommendation," *ACM Trans. Inform. Syst. (TOIS)*, vol. 40, no. 2, 2021.

[21] Y. Tao, Y. Li, S. Zhang, Z. Hou, and Z. Wu, "Revisiting graph based social recommendation: A distillation enhanced social graph network," in *Proc. ACM Web Conf. (WWW)*, 2022, pp. 2830–2838.

[22] L. Xia, Y. Shao, C. Huang, Y. Xu, H. Xu, and J. Pei, "Disentangled graph social recommendation," in *IEEE 39th Int. Conf. Data Eng. (ICDE)*, Anaheim, CA, USA, Apr. 2023, pp. 2332–2344.

[23] J. Yu, H. Yin, J. Li, Q. Wang, N. Q. V. Hung, and X. Zhang, "Self-supervised multi-channel hypergraph convolutional network for social recommendation," in *Proc. Web Conf (WWW)*, 2021, pp. 413–424.

[24] J. Yu, H. Yin, M. Gao, X. Xia, X. Zhang, and N. Q. Viet Hung, "Socially-aware self-supervised tri-training for recommendation," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining (KDD)*, 2021, pp. 2084–2092.

[25] Y. Quan, J. Ding, C. Gao, L. Yi, D. Jin, and Y. Li, "Robust preference-guided denoising for graph based social recommendation," in *Proc. ACM Web Conf. 2023 (WWW)*, 2023, pp. 1097–1108.

[26] T. Wang, L. Xia, and C. Huang, "Denoised self-augmented learning for social recommendation," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2023.

[27] M. Zhao et al., "Bilateral filtering graph convolutional network for multi-relational social recommendation in the power-law networks," *ACM Trans. Inf. Syst. (TOIS)*, vol. 40, no. 2, pp. 1–24, 2021.

[28] L. Chuanzhen, X. Han, C. Juanjuan, and W. Hui, "Social recommender systems using collaborative user network embedding with bias," in *Proc. IEEE 5th Int. Conf. Comput. Commun. (ICCC)*. Piscataway, NJ, USA: IEEE Press, 2019, pp. 230–234.

[29] H. Li, L. Li, G. Xv, C. Lin, K. Li, and B. Jiang, "Spex: A generic framework for enhancing neural social recommendation," *ACM Trans. Inf. Syst. (TOIS)*, vol. 40, no. 2, 2021.

[30] X. Chen, P. I. Lei, Y. Sheng, Y. Liu, and Z. Gong, "Social influence learning for recommendation systems," in *Proc. 33rd ACM Int. Conf. Inf. Knowl. Manage.*, 2024, pp. 312–322.

[31] C. Zhang and G. Li, "Biasrec: A general bias-aware social recommendation model," in *Proc. Int. Conf. Database Syst. Adv. Appl.* Cham, Switzerland: Springer, 2024, pp. 101–116.

[32] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," presented at the NIPS Deep Learning and Representation Learning Workshop, vol. 14, no. 7, 2015, pp. 38–39.

[33] Z. Liu and L. Zhang, "Enhanced self-supervised graph learning algorithm for social recommendation," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*. Piscataway, NJ, USA: IEEE Press, 2024, pp. 1–8.

[34] Y. Yang, L. Wu, Z. Wang, Z. He, R. Hong, and M. Wang, "Graph bottlenecked social recommendation," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2024, pp. 3853–3862.

[35] C. Wei, C. Hu, C.-D. Wang, and S. Huang, "Time-aware multibehavior contrastive learning for social recommendation," *IEEE Trans. Ind. Inform.*, vol. 20, no. 4, pp. 6424–6435, Apr. 2024.

[36] X. Liu et al., "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, 2023, pp. 857–876.

[37] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[38] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," Jul. 2018, *arXiv:1807.03748*.

[39] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*. JMLR.org, 2020.

[40] J. Qiu et al., "GCC: Graph contrastive coding for graph neural network pre-training," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2020, pp. 1150–1160.

[41] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Advances in Neural Information* Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Red Hook, NY, USA: Curran Associates, Inc., 2020, pp. 5812–5823.

[42] T. Zang, Y. Zhu, R. Zhang, C. Wang, K. Wang, and J. Yu, "Contrastive multi-view interest learning for cross-domain sequential recommendation," *ACM Trans. Inf. Syst.*, vol. 42, no. 3, pp. 1–28, 2023.

[43] L. Wu, H. Lin, Z. Gao, C. Tan, and S. Z. Li, "Self-supervised on graphs: Contrastive, generative, or predictive," vol. abs/2105.07342, 2021.

[44] J. Wu et al., "Self-supervised graph learning for recommendation," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval (SIGIR)*, 2021, pp. 726–735.

[45] J. Yu et al., "Are graph augmentations necessary? simple graph contrastive & nbsp;learning for recommendation," in *Proc. 45th*

Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval (SIGIR), 2022, pp. 1294–1303.
[46] Z. Lin, C. Tian, Y. Hou, and W. X. Zhao, "Improving graph collaborative filtering with neighborhood-enriched contrastive learning," in *Proc. ACM Web Conf. (WWW)*, 2022, pp. 2320–2329.
[47] X. Cai, C. Huang, L. Xia, and X. Ren, "LightGCL: Simple yet effective graph contrastive learning for recommendation," in *11th Int. Conf. Learn. Representations (ICLR)*, 2023.
[48] J. Zhang, M. Gao, J. Yu, L. Guo, J. Li, and H. Yin, "Double-scale self-supervised hypergraph learning for group recommendation," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2021, pp. 2557–2567.
[49] C. Wang, Y. Liang, Z. Liu, T. Zhang, and P. S. Yu, "Pre-training graph neural network for cross domain recommendation," in *Proc. IEEE 3rd Int. Conf. Cogn. Mach. Intell. (CogMI)*, 2021, pp. 140–145.
[50] X. Xie et al., "Contrastive learning for sequential recommendation," in *IEEE 38th Int. Conf. Data Eng. (ICDE)*, Kuala Lumpur, Malaysia, May 2022, pp. 1259–1273.
[51] X. Xia, H. Yin, J. Yu, Q. Wang, L. Cui, and X. Zhang, "Self-supervised hypergraph convolutional networks for session-based recommendation," *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 5, 2021, pp. 4503–4511.
[52] J. Wu, W. Fan, J. Chen, S. Liu, Q. Li, and K. Tang, "Disentangled contrastive learning for social recommendation," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2022, pp. 4570–4574.
[53] W. Jin, Z. Zhao, P. Zhang, J. Zhu, X. He, and Y. Zhuang, "Hierarchical cross-modal graph consistency learning for video-text retrieval," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval (SIGIR)*, 2021, pp. 1114–1124.
[54] Y. Yang, C. Huang, L. Xia, and C. Li, "Knowledge graph contrastive learning for recommendation," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval (SIGIR)*, 2022, pp. 1434–1443.
[55] X. Wang, W. Lu, M. Ester, C. Wang, and C. Chen, "Social recommendation with strong and weak ties," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2016, pp. 5–14.
[56] Y. Sun, D. Zhu, H. Du, and Z. Tian, "Motifs-based recommender system via hypergraph convolution and contrastive learning," *Neurocomputing*, vol. 512, pp. 323–338, Nov. 2022.
[57] J. Yu, H. Yin, X. Xia, T. Chen, J. Li, and Z. Huang, "Self-supervised learning for recommender systems: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 1, pp. 335–355, 2024.

Shixiao Yang received the bachelor's degree in data science and big data technology, from Beijing Institute of Technology, Beijing, China, in 2023, where he is currently working toward the master's degree in computer science and technology.

His research interests include recommender systems and airspace intelligent networking.

Zhida Qin received the B.S. degree in electronic and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 2014, and the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China.

Currently, he is an Assistant Professor with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. His research interests include recommendation systems, graph neural networks, and online learning.

Enjun Du is currently an undergraduate student with the Department of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China.

His research interests include graph deep learning, large language models (LLM), and recommendation systems.

Pengzhan Zhou (Member, IEEE) received the B.S. degree in mathematics and applied mathematics from Shanghai Jiao Tong University, Shanghai, China, in 2014, and the Ph.D. degree in electrical engineering from Stony Brook University, NY, USA, in 2020.

He is a Professor with the College of Computer Science, Chongqing University, Chongqing, China. His research interests include machine learning, wireless sensor networks, and performance evaluation of network protocols and algorithms.

Tianyu Huang received the B.E. degree in computer science from Jilin University, Changchun, China, in 2002, and the Ph.D. degree in computer science from Beijing Institute of Technology, Beijing, China, in 2007.

She joined as the faculty with Beijing Institute of Technology, in 2007. She worked as a Visiting Scholar with the Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA, from 2012 to 2013. She is a Professor with the School of Computer Science and Technology, Beijing Institute of Technology. Her research interests include human motion modeling and simulation, crowds simulation, and virtual reality.