Socially Enhanced Adaptive Contrastive Learning GNN Recommendation(SDACGR) for Telecom Operators Users Behaviours Prediction

Anonymous ACL submission

Abstract

In the era of information technology, user behavior prediction systems have become increasingly important for telecom operators such as China Mobile, enabling personalized services and improving user experience. However, existing GNN-based recommendation methods still suffer from data sparsity and cold-start issues, while often ignoring the varying importance of users and items. To address these challenges, this paper proposes a socially enhanced adaptive contrastive graph recommendation model, termed SDACGR. Specifically, social information is incorporated to alleviate data sparsity, and a social denoising module is designed to remove noisy social relations with low interaction similarity. The user representations learned from interaction and social views are then adaptively fused for recommendation. In addition, to better handle complex graph structures, we introduce an adaptive contrastive view generation strategy based on a multi-head graph attention network, which generates multiple enhanced views for contrastive learning. Experimental results on several widely used collaborative filtering and social recommendation datasets demonstrate that SDACGR consistently outperforms strong baselines. Further ablation and parameter studies verify the effectiveness and robustness of each component.

1 Introduction

With the continuous expansion of mobile operators' services, subscriber behavior prediction systems have become increasingly important for enabling personalized customer service and improving user experience and content distribution efficiency. Traditional methods, such as collaborative filtering(Goldberg et al., 1992) and matrix decomposition(Koren et al., 2009), rely mainly on shallow user-item interaction data and struggle to capture complex interactions and higher-order graph structures, such as user-social relationships and item-attribute associations, resulting in limited personalization and interpretability. With the integration of graph theory and deep learning, graph neural network (GNN)-based prediction methods(Scarselli et al., 2008) have achieved significant progress due to their strong structural modeling capability. However, GNN-based recommendation still faces the following challenges:

1. Mainstream graph convolutional networks (Fan et al., 2019; Song et al., 2019) aggregate messages without distinguishing the importance of users and items, since they only weight neighbors by node degree.

2. The user-item interaction matrix is highly sparse. Compared with the vast number of items, which may reach hundreds of millions, each user typically interacts with only a few dozen items, making it difficult for recommendation models to learn effective user preferences and thereby affecting prediction performance.

3. The cold-start problem remains severe: insufficient historical behavior data for new users makes accurate prediction difficult.

To address these issues, this paper proposes a socially denoised adaptive contrastive graph recommendation method, SDACGR. To mitigate social noise, we design a social denoising module. Specifically, user-item interaction information is used for graph convolution to derive user embeddings, which are then used to compute user similarity. Social relationships with low similarity are filtered out, and graph convolution is subsequently performed on the social network. The user embeddings learned from the interaction graph and the social graph are then fused for final recommendation prediction. This mechanism removes

noisy social signals while preserving socially useful information, thereby improving recommendation performance.

Contrastive learning has attracted considerable attention in many fields. By leveraging data augmentation, noise injection, and increased training difficulty, it encourages the model to capture essential data characteristics, thereby improving representation quality and robustness. To address the complexity of graph neural networks, we further propose an adaptive contrastive learning framework for generating contrastive views. Specifically, we introduce an adaptive view augmentation method based on graph attention networks to generate high-quality contrastive views for alleviating data sparsity. A multi-head graph attention network is used to generate multiple distinct contrastive views for user and item nodes, which are then employed in contrastive learning to enhance robustness and generalization. Extensive experiments on three social recommendation datasets demonstrate that SDACGR outperforms strong baseline recommendation methods. In addition, ablation and parameter studies further verify the effectiveness of each module.

2 Related Works

2.1 Social Recommendation

Social networks model users as nodes and their social relationships (e.g., friendships or followings) as edges. In social recommendation, such networks are used to improve personalized recommendation by complementing user-item interaction data with social relations, thereby better capturing users' interests and preferences. Numerous methods have been proposed to leverage social information for alleviating data sparsity and improving recommendation performance. Representative examples include TrustMF(Yang et al., 2016) based on matrix factorization, DiffNet(Wei et al., 2022), DGREC(Song et al., 2019), and NGCF+(Wang et al., 2019) based on graph convolutional neural networks, EATNN(Chen et al., 2019) based on attention mechanisms, and MHCH(Yu et al., 2021b) and SEPT(Yu et al., 2021a) based on self-supervised learning.

However, social relations often contain noise and may not fully reflect users' true interests. Irrelevant social connections can obscure preference patterns and reduce recommendation quality. In addition, social information is semantically different from user behavior data. Relying solely on social relations may cause models to overemphasize superficial connections while ignoring actual interaction patterns, leading to overfitting or inaccurate preference learning. For example, methods such as NGCF, which do not filter social noise, may perform worse than methods that do not use social information.

2.2 Contrastive Learning

Contrastive learning is a self-supervised learning approach that learns discriminative representations by constructing positive pairs of similar samples and negative pairs of dissimilar samples. Its objective is to bring similar samples closer in the feature space while pushing dissimilar samples apart, thereby improving representation quality and generalization. In recommendation, Wu et al. (2021) proposed self-supervised graph learning (SGL), which generates different graph views through node dropout, edge dropout, and random-walk subgraph sampling, and then performs contrastive learning on these views. Lin et al. (2022) proposed NCL, which constructs structural neighbors and semantic neighbors from node representations to extract self-supervised signals and improve graph recommendation performance.

Contrastive learning with self-supervised signals has become an effective solution for data sparsity and noise. In graph collaborative filtering, existing methods usually maximize consistency across different views by learning multiple representations of the same node. For example, SGL(Wu et al., 2021) applies node dropout, edge dropout, and random walks to randomly generate multiple contrastive views of the original graph. However, such random augmentation may damage graph structure and node features. Preserving the original topology is important because it contains key structural information, especially for GCN-based models that rely on graph connectivity.

3 Methodology

3.1 Problem Formulation

In a recommendation system, we consider M users and N items, denoted as $U = \{u_1, u_2, u_3, \ldots, u_m\}$ and $I = \{i_1, i_2, i_3, \ldots, i_n\}$, respectively. The user-item interaction matrix $R \in R^{M \times N}$ records user-item interactions, where $R_{u,i} = 1$ indicates that user u has interacted with

item i, and $R_{u,i} = 0$ otherwise.

For social relations, we define the social matrix $R_s \in R^{M \times M}$, where $R_{s(u,v)} = 1$ indicates a social connection between users u and v, and $R_{s(u,v)} = 0$ otherwise. As described above, user-item interactions and social relations can be represented as a user-item interaction graph and a social graph, denoted as $G_i = \{V_{ui}, E_i\}$ and $G_s = \{V_u, E_u\}$, respectively. Based on these inputs, the social recommendation task aims to capture the structural associations in the two graphs and learn the embeddings Emb_u and Emb_i for users and items. The model then predicts the interaction probability between user u and all items i, and recommends the top-k items with the highest scores for each user.

3.2 Description of SDACGR Method

The overall framework of SDACGR is shown in Fig. 1. This section focuses on the key components of social enhancement and self-supervised learning, including social information integration, social noise filtering, and adaptive contrastive view generation.

3.2.1 Initialization

We first randomly initialize the user embedding matrix $Emb_u \in R^{M \times d}$ and the item embedding matrix $Emb_i \in R^{N \times d}$. Specifically, they are represented as:

$$Emb_u = [Emb_{u1}, Emb_{u2}, \ldots, Emb_{uM}] \quad (1)$$

$$Emb_i = [Emb_{i1}, Emb_{i2}, \ldots, Emb_{iN}] \quad (2)$$

Here, d denotes the embedding dimension, typically set to 64 or 128. Learning informative initial representations is the objective of model training. The user and item embeddings are then concatenated to form $Emb \in R^{(M+N) \times d}$, enabling efficient graph convolution via matrix multiplication:

$$Emb = [Emb_{u1}, \ldots, Emb_{uM}, Emb_{i1}, \ldots, Emb_{iN}] \quad (3)$$

We initialize the user-item interaction matrix $R \in R^{M \times N}$ and the normalized user-item interaction matrix $S_r \in R^{(M+N) \times (M+N)}$. In addition, based on user social information, we construct the initial social matrix $R_s \in R^{M \times M}$ and normalize it as $S_{social} = D^{-\frac{1}{2}}(R_{social} + I)D^{-\frac{1}{2}}$, where I is the identity matrix and D is the diagonal degree matrix.

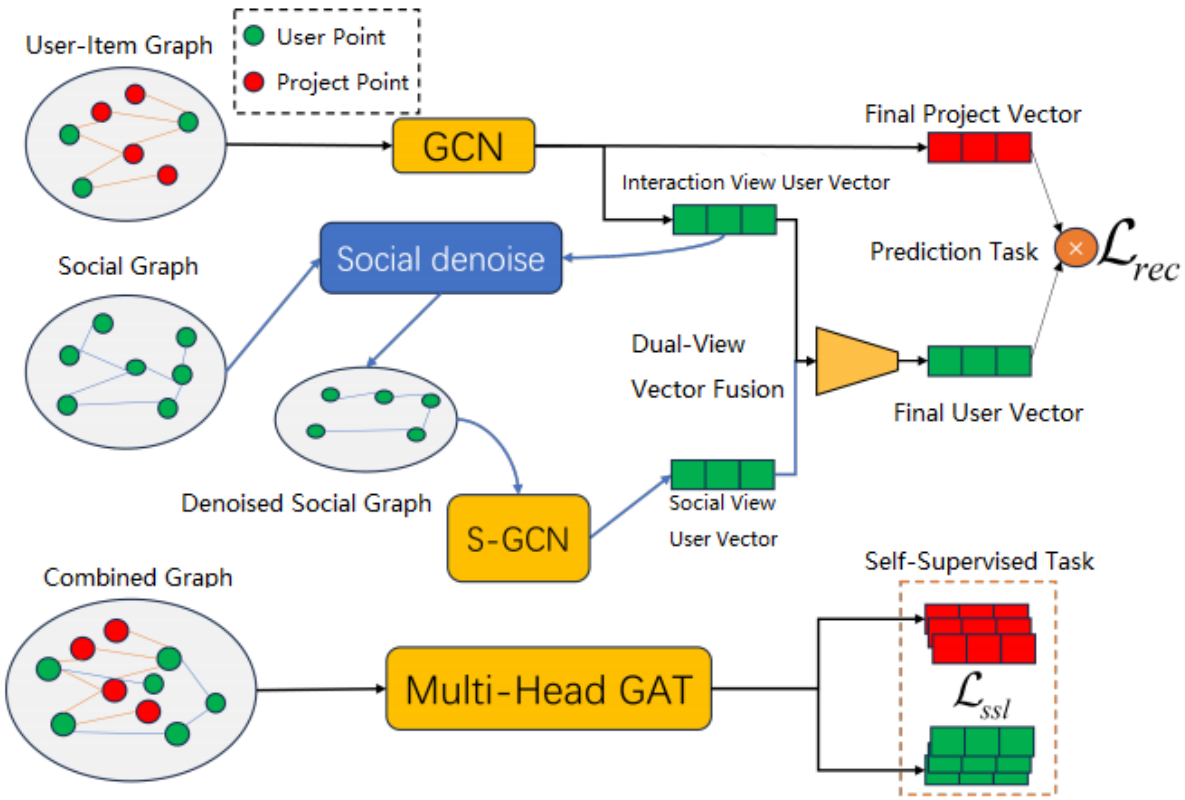


Figure 1: SDACGR model architecture diagram

3.2.2 User-Project Interaction Learning

The final user and item representations are learned from the user-item interaction graph. We adopt LightGCN as the graph encoder, which performs efficient neighborhood aggregation through weighted summation:

$$e_u^{(k+1)} = \sum_{i \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} e_i^{(k)} \tag{4}$$

$$e_i^{(k+1)} = \sum_{u \in N_i} \frac{1}{\sqrt{|N_i|}\sqrt{|N_u|}} e_u^{(k)} \tag{5}$$

Here, N_u and N_i denote the neighbors of user node u and item node i, respectively, and k indicates the k-th graph convolution layer. This process can also be implemented efficiently in matrix form:

$$E^{(0)} = Emb_u \oplus Emb_i \tag{6}$$

$$E^{(k+1)} = S_r E_r^{(k)} \tag{7}$$

After multiple graph convolution layers, we obtain the representations at each layer and compute the final interaction-view embeddings by averaging:

$$E = \frac{1}{k} \sum_{l=1}^{k} E^{(l)} \tag{8}$$

$$E_u = \{E_a | a \in [1, m]\}, \quad E_i = \{E_a | a \in (m, m{+}n]\} \tag{9}$$

The resulting item embeddings are directly used for prediction, whereas the user embeddings are further refined by social-view learning and then fused with the social representations to obtain the final user embeddings.

3.2.3 Social Information Denoising

In online platforms, social relations such as family, friends, and colleagues are often passively formed and may introduce noise into recommendation. To mitigate this issue, we filter out social edges that do not contribute to recommendation performance. Specifically, we use the interaction-view representations to guide social propagation by measuring user similarity in the interaction space. If the similarity between two users falls below a threshold, their social link is removed. The overall process is illustrated in Fig. 1.

For the user vectors $\left(e_i^r, e_j^r\right)$ obtained from the interaction view, the similarity between user pair (i, j) is defined as $sim_{i,j}^r = f_r(e_i^r, e_j^r)$. The similarity projection function $f_r(\bullet, \bullet)$ maps user vectors from the interaction view into a latent embedding space for alignment and similarity estimation, as shown in Fig. 2.

The two input vectors are first projected into the latent space through a shared weight matrix and bias, followed by a ReLU activation:

$$h_u = \sigma(W_1 e_u + b_1) \tag{10}$$

$$h_v = \sigma(W_1 e_v + b_1) \tag{11}$$

Here, $\sigma(\bullet)$ denotes the LeakyReLU activation function; $W_1 \in R^{h \times d}$ is the shared weight matrix, where d is the input dimension and h is the hidden dimension; and $b_1 \in R^h$ is the shared bias.

Next, we compute feature interactions by taking the absolute difference and element-wise product of the two vectors, and then concatenate them:

$$diff = |h_u - h_v| \tag{12}$$

$$prod = h_u \odot h_v \tag{13}$$

$$c = (diff \oplus prod) \in R^{2h} \tag{14}$$

Here, $\odot$ denotes the Hadamard product, $|\cdot|$ denotes the absolute value, and $\oplus$ denotes vector concatenation.

Finally, the similarity score is obtained by progressively reducing the dimensionality through fully connected layers:

$$g = \sigma(W_2 c + b_2) \tag{15}$$

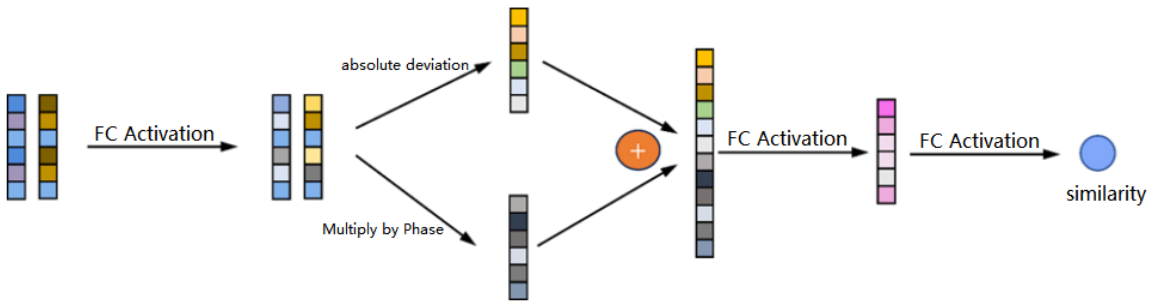


Figure 2: Similarity computation process diagram

$$s = S(W_3 g + b_3) \tag{16}$$

where $W_2 \in R^{32\times 2h}$, $b_2 \in R^{32}$, $W_3 \in R^{1\times 32}$, and $b_3 \in R^1$ are learnable parameters; $\sigma(\bullet)$ denotes the LeakyReLU activation function, and $S(\bullet)$ denotes the Sigmoid activation function.

For the original social graph G_s, the node set is V_u and the edge set is E_s. After learning from interaction information, we obtain user representations and compute interaction-based user similarity using the module $sim_{u,v}$, where $sim_{u,v}$ denotes the similarity between users u and v. We then denoise the original graph by removing edges whose similarity is below the threshold, obtaining G'_s. The node set remains V_u, while the new edge set is E'_s, defined as:

$$E'_s = \{(u,v)|(u,v) \in E, sim_{u,v} \geq \beta\} \tag{17}$$

Here, β is the denoising threshold. Only social relations with similarity greater than β are retained. In experiments, β is adjusted dynamically by setting the denoising ratio. After obtaining the denoised social graph, we perform social graph learning to obtain social-based user representations, which are subsequently fused with interaction-based representations for recommendation.

3.2.4 Social Learning and Fusion of User Vectors

After obtaining the denoised user social graph, we perform social-view learning using the following graph convolution process:

$$E_s^{(0)} = Emb_u \tag{18}$$

$$E_s^{(k+1)} = (S'_{social})E_s^{(k)} \tag{19}$$

The interaction view and the social view produce two distinct user representations, E_r and E_s, capturing complementary semantic information. We adopt an adaptive fusion strategy. Specifically, the representations from the two views are first concatenated into a higher-dimensional vector, from which the importance of the social view is estimated. The final user representation is then computed as a weighted sum of the two views:

$$E_{rs} = E_r \oplus E_s \tag{20}$$

$$\alpha = \lambda \times \sigma(W_1 E_{rs} + b) \tag{21}$$

$$E_u = \alpha E_s + (1-\alpha)E_r \tag{22}$$

Here, λ is a hyperparameter controlling the maximum contribution of the social view, and $\sigma(\bullet)$ is the sigmoid function that maps values to $(0,1)$.

3.2.5 Adaptive Comparison View Generation

After incorporating social information, the model becomes more structurally complex, making fixed contrastive view generation less adaptable to different graph architectures. To address this issue, we propose an adaptive contrastive view generation strategy that dynamically adjusts according to the structural characteristics of the graph neural network. Specifically, we introduce a graph attention mechanism that leverages both node attributes and graph structure to generate multiple contrastive views. Inspired by NCLA in graph node classification [56], we employ a multi-head graph attention network (Multi-Head GAT [41]) to generate k learnable graph-enhanced views.

First, we reconstruct the graph structure $G_A = \{V, E\}$ for contrastive view generation, where V denotes the set of nodes, including both users and items, and E denotes the edge set. Notably, E includes not only user-item interaction edges but also user-user social edges. The corresponding adjacency matrix $A \in \{0,1\}^{(M+N)\times(M+N))}$ is defined as follows: $A_{i,j} = 1$ if nodes i and j are neighbors, and $A_{i,j} = 0$ otherwise. The node embeddings are denoted as $Emb \in R^{(M+N))\times d}$.

Given Emb and A, the adaptive enhancement module generates k learnable topological views $\{Emb, \tilde{A}^{(k)}\}$. Although each view shares the same input embeddings, the learned relation parameters differ, leading to different outcomes in subsequent convolution operations. The framework is shown in Fig. 3.

For the k-th view with adjacency matrix $\tilde{A}^{(k)}$, if two nodes i and j are connected:

$$\tilde{A}_{i,j}^{(k)} = \frac{e^{LeakyReLU(\varphi^{(k)}(W^{(k)}e_i \oplus W^{(k)}e_j))}}{\sum_{v\in N_i} e^{LeakyReLU(\varphi^{(k)}(W^{(k)}e_i \oplus W^{(k)}e_v))}} \tag{23}$$

Here, $\varphi^{(k)} \in R^{1\times 2d}$ denotes the learnable parameters of the k-th view, $W^{(k)}$ is the feature transformation matrix for the k-th view, and N_i is the neighborhood of node i.

If nodes i and j are not connected, then $\tilde{A}_{i,j}^{(k)} = 0$. After obtaining the enhanced matrix $\tilde{A}_{i,j}^{(k)}$, graph convolution is performed to obtain the final node representations:

$$h_i^{(k)} = ELU\left(\sum_{j\in N_i} \hat{A}_{i,j}^{(k)} W^{(k)} e_j\right) \tag{24}$$

where $h_i^{(k)}$ is the representation of node i in the k-th contrastive view.

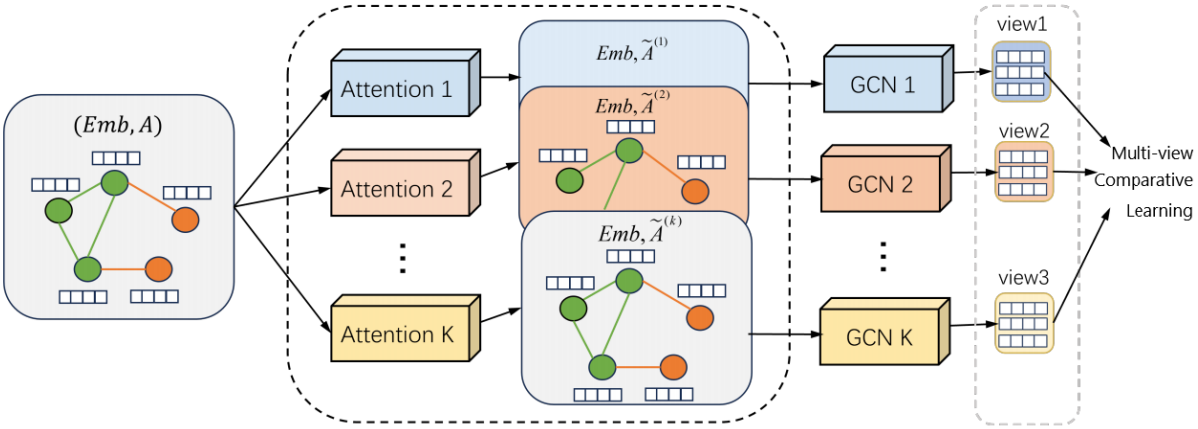


Figure 3: Example of adaptive view generation framework

After obtaining the enhanced views, contrastive learning is applied to further refine the representations of users and items. Since the multi-head graph attention network generates k enhanced views, multiple contrastive learning objectives can be computed accordingly.

3.2.6 Loss Function

The overall training objective consists of a supervised ranking task and a self-supervised contrastive learning task. For the main task, we adopt Bayesian Personalized Ranking (BPR) loss:

$$L_{bpr} = -\sum_{(u,i,j)\in D} \log(\sigma(e_u^T e_i - e_u^T e_j)) \quad (25)$$

Here, $u \in U$ denotes the set of users; $i \in I_u^+$ denotes items that user u has interacted with as positive samples; $i \in I_u^-$ denotes items that user u has not interacted with as negative samples; and (u, i, j) is a triplet where user u should prefer item i over item j. σ is the sigmoid activation function, which maps the score difference to $(0, 1)$.

For the self-supervised task, we use InfoNCE loss:

$$L_{ssl} = -\log \frac{\exp(sim(q, k^+)/\tau)}{\sum_{i=0}^{N} \exp(sim(q, k_i)/\tau)} \quad (26)$$

where q denotes the query sample, k^+ is the positive sample corresponding to q, k_i are negative samples, $sim(\cdot)$ measures the similarity between two nodes, and τ is the temperature coefficient controlling the smoothness of the contrastive distribution. Since k augmented views are generated, multiple contrastive losses are computed and averaged.

The model jointly exploits supervised signals from user-item interaction history and social relations, as well as self-supervised signals from contrastive learning. The final objective is:

$$L = L_{bpr} + \lambda L_{ssl} \quad (27)$$

where L_{bpr} is the pairwise ranking loss, L_{ssl} is the self-supervised loss, and λ is the weight of the contrastive learning term.

4 Experimental Results and Analysis

We evaluate SDACGR on three datasets derived from China Mobile, including consumption, rating, and music scenarios. Due to privacy constraints, equivalent datasets with identical formats are released. This section presents the datasets, evaluation metrics, baseline methods, and experimental results, including performance comparison, ablation studies, and parameter analysis.

- Dataset: Ciao, Yelp2019, and LastFM.
- Evaluation Metrics: We use Recall@10 and NDCG@10 as evaluation metrics.
- Baselines: We compare SDACGR with representative recommendation models, including SBPR(Zhao et al., 2014), Diffnet++(Wu et al., 2019), LightGCN(He et al., 2020), SocialLGN(Liao et al., 2022), MHCH(Yu et al., 2021b), and SEPT(Yu et al., 2021a).
- Experimental Settings: All baselines are implemented following their original papers, and hyperparameters are carefully tuned for optimal performance.

Table 1: Performance comparison across different models and datasets.

Model	Ciao		Yelp2019		LastFM	
	R@10	N@10	R@10	N@10	R@10	N@10
SBPR	0.02588	0.02683	0.05473	0.04320	0.17245	0.22310
Diffnet++	0.03579	0.03722	0.06314	0.04816	0.18775	0.22412
LightGCN	0.04103	0.04239	0.06534	0.04977	0.19486	0.23388
SocialLGN	0.04055	0.04498	0.06936	0.05141	0.20263	0.23401
MHCH	0.04238	0.04513	0.06951	0.05554	0.19638	0.23500
SEPT	0.04391	0.04649	0.07177	0.05734	0.20942	0.24503
SDACGR	**0.04540**	**0.04828**	**0.07671**	**0.05831**	**0.20883**	**0.25018**
Improve	3.39%	3.85%	6.88%	6.19%	1.90%	2.11%

4.1 Performance Comparison Experiment

Table 1 reports the performance of SDACGR and competing methods on the three datasets. SDACGR consistently achieves the best Recall@10 and NDCG@10 scores, demonstrating the effectiveness of integrating social denoising and adaptive contrastive learning.

The improvement on LastFM is relatively modest (approximately 2%). This can be attributed to two factors: (1) LastFM contains relatively dense user-item interactions, which already provide rich higher-order structural information; and (2) the number of social relations is limited, reducing the contribution of social information.

In contrast, SDACGR achieves gains exceeding 6% on Yelp2019. Since Yelp2019 contains substantially more social relations than interaction records, recommendation performance relies more heavily on effective utilization of social information. The proposed social denoising module helps filter noisy user-user relations and improves the quality of social information propagation.

4.2 Ablation Experiment

To evaluate the contribution of each component, we consider the following variants:

1. SDACGR-S: Removes social information and uses only user-item interactions.

2. SDACGR-D: Removes the social denoising module and retains the original social graph.

3. SDACGR-CL: Removes the contrastive learning module.

Table 2: Ablation experiment

Model	Ciao		Yelp2019		LastFM	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
SDACGR-S	0.04233	0.04447	0.07215	0.05494	0.19842	0.23575
SDACGR-D	0.04383	0.04744	0.07328	0.05515	0.19738	0.23315
SDACGR-CL	0.04198	0.04413	0.07536	0.05734	0.19612	0.23576
SDACGR	**0.04540**	**0.04828**	**0.07671**	**0.05831**	**0.20883**	**0.25018**

Table 2 presents the ablation results. Removing any component consistently degrades performance across all datasets, indicating that each module contributes to the final recommendation quality.

Specifically, removing social information (SDACGR-S) leads to notable performance drops, highlighting the importance of social signals. Eliminating social denoising (SDACGR-D) also reduces performance, confirming that filtering noisy social relations improves recommendation effectiveness. Furthermore, the degradation observed in SDACGR-CL demonstrates the benefit of adaptive contrastive learning in learning robust representations.

5 Conclusion

In the era of rapidly growing data volumes, recommendation systems have become essential for alleviating information overload and providing personalized content services, making them an important research topic in both academia and industry. This paper presents an in-depth study of graph neural network-based recommendation methods.

To address data sparsity in recommendation systems, we propose the Socially Denoised Adaptive Contrastive Graph Recommendation (SDACGR) model. Specifically, a social denoising module is designed to filter noisy social relations using user embeddings learned from user-item interaction views. Social edges with low interaction similarity are removed, and the resulting denoised social graph is used to learn social-aware user embeddings. These embeddings are then adaptively fused with interaction-based user embeddings for recommendation. In addition, we propose an adaptive contrastive learning framework for complex graph neural networks. By leveraging GAT to generate multiple informative contrastive views, the model further alleviates data sparsity and improves representation learning.

Extensive experiments on multiple datasets demonstrate that SDACGR consistently outperforms state-of-the-art recommendation methods, verifying the effectiveness of incorporating social information into recommendation. The results further highlight the importance of social denoising and self-supervised contrastive learning for improving recommendation performance under sparse settings.

Limitations

Although SDACGR achieves consistent improvements over strong baselines across multiple benchmark datasets, several limitations remain.

1. The proposed framework is built on static interaction and social graphs. In real-world recommendation systems, both user-item interactions and social relationships evolve continuously over time. Consequently, the current model may not fully capture temporal preference shifts or dynamic social influence patterns. Extending the proposed social denoising and adaptive contrastive learning framework to dynamic graph settings is an important direction for future research.

2. SDACGR relies primarily on collaborative signals derived from user-item interactions and social relations, without incorporating multimodal information. Modern online platforms often provide rich auxiliary content, such as textual reviews, images, audio, and videos, which may contain valuable semantic information beyond graph structures. Integrating multimodal feature representations into the current framework could further improve recommendation quality, particularly in sparse-interaction scenarios.

3. The effectiveness of the proposed social enhancement mechanism depends on the availability and quality of social relations. As observed in the experiments, SDACGR achieves larger improvements on datasets with relatively rich social connections (e.g., Yelp2019), while the gains are more limited on datasets containing fewer social links (e.g., LastFM). Therefore, the benefits of social-aware modeling may vary across application scenarios with different social network densities.

4. The adaptive contrastive learning module introduces additional computational overhead compared with standard graph-based recommendation models. Specifically, generating multiple augmented views through multi-head graph attention networks increases both memory consumption and training complexity. Although the experimental results show that a small number of views (e.g., $K = 2$ or 3) achieves a favorable trade-off between effectiveness and efficiency, scalability on extremely large-scale recommendation systems requires further investigation.

5. The model contains several hyperparameters, including the denoising ratio, the number of contrastive views, and the graph propagation depth. Experimental results demonstrate that these factors influence performance to varying degrees. While the proposed model exhibits stable improvements across datasets, more adaptive or automated hyperparameter optimization strategies may further enhance robustness and generalization.

Acknowledgments

References

Chong Chen, Min Zhang, Chenyang Wang, Weizhi Ma, Minming Li, Yiqun Liu, and Shaoping Ma. 2019. An efficient adaptive transfer neural network for social-aware recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 225–234. Association for Computing Machinery.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *Proceedings of the 2019 World Wide Web Conference*, pages 417–426. Association for Computing Machinery.

David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. 1992. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 639–648. Association for Computing Machinery.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Jie Liao, Wei Zhou, Fengji Luo, Junhao Wen, Min Gao, Xiuhua Li, and Jun Zeng. 2022. SocialLGN: Light graph convolution network for social recommendation. *Information Sciences*, 589:595–607.

Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM Web Conference 2022*, pages 2320–2329. Association for Computing Machinery.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based social recommendation via dynamic graph attention networks. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 555–563. Association for Computing Machinery.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 165–174. Association for Computing Machinery.

Wei Wei, Chao Huang, Lianghao Xia, Yong Xu, Jiashu Zhao, and Dawei Yin. 2022. Contrastive meta learning with behavior multiplicity for recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1120–1128. Association for Computing Machinery.

Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 726–735. Association for Computing Machinery.

Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 235–244. Association for Computing Machinery.

Bo Yang, Yu Lei, Jiming Liu, and Wenjie Li. 2016. Social collaborative filtering by trust. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(8):1633–1647.

Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. 2021a. Socially-aware self-supervised tri-training for recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2084–2092. Association for Computing Machinery.

Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021b. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *Proceedings of the Web Conference 2021*, pages 413–424. Association for Computing Machinery.

Tong Zhao, Julian McAuley, and Irwin King. 2014. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 261–270. Association for Computing Machinery.

A Dataset Details

- Ciao: The Ciao dataset is sourced from the website http://www.ciao.co.uk, which includes information on items purchased by users and their ratings, as well as social network data from users' trusted friends.
- Yelp2019: The Yelp dataset was released by the Yelp platform in 2019. Yelp is a well-known local business review platform where users can browse merchant information, rate businesses, and the platform also records social relationships between users.
- lastFM: Last.fm is a globally renowned music platform where users can record the songs they listen to, follow other users, establish friend connections, and share music preferences within their social circles.

The three datasets used in the experiment all originate from well-known mainstream platforms, featuring a broad user base and high credibility. These datasets cover different fields, vary in scale and density, but all contain authentic and reliable data, making the results of this experiments practically valuable. Key statistical information for the three datasets is shown in Table 3.

Table 3: Dataset Statistics

Statistics	Ciao	Yelp2019	LastFM
Number of users	6,672	19,539	1,892
Number of items	98,875	21,266	17,632
Number of interactions	198,181	450,884	92,834
Number of social relationships	109,503	864,157	25,434
Data density	0.03%	0.11%	0.28%

B Evaluation Indicators

The primary objective of a recommendation system is to generate personalized recommendation lists for users. Consequently, the evaluation metrics must simultaneously consider two aspects: the accuracy of the recommended items and the rationality of the recommendation rankings. Specifically, the recommendation model must not only accurately predict user interest in a particular item (i.e., identify positive samples) but also discern the degree of preference for different items, ensuring that more preferred items are ranked higher in the recommendation list. To comprehensively assess the recommendation system enhancement algorithm presented in this paper, we referred to the widely accepted evaluation framework in the field

and drew inspiration from seminal works such as NGCF, LightGCN, and SGL, selecting the following two key metrics:

1. Recall: The ratio of positive instances in the predicted outcomes to all actual positive instances, used to gauge the recommendation list's ability to encompass items of interest to users, reflecting the coverage and precision of the recommendations, and serving as a crucial metric for assessing the system's recall capability. The formula is as follows:

$$Recall@K = \frac{TP}{TP + FN} \tag{28}$$

   K—length of the recommendation list; TP—true positives, the number of positive samples among the top recommendations; FN—false negatives, i.e., the number of items the user is genuinely interested in but were omitted; TP+FN—represents all positive samples for the user.

2. NDCG (Normalized Discounted Cumulative Gain): Utilized to assess the ranking quality of items in the recommendation list, it strongly emphasizes whether more preferred items are given priority in the recommendations. Firstly, compute the discounted cumulative gain (DCG):

$$DCG@K = \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{29}$$

   rel_i—the relevance score of the i-th item. IDCG denotes the DCG under ideal conditions, recalculated post sorting the items of interest to users in the test set by relevance from high to low. Ultimately, compare DCG with IDCG to compute NDCG:

$$NDCG@K = \frac{DCG@K}{IDCG@K} \tag{30}$$

C Baseline

1. SBPR(Zhao et al., 2014): A recommendation model based on matrix factorization, integrating social relationship information with BPR to enhance the precision of personalized rankings.

2. Diffnet++(Wu et al., 2019): A graph-based social recommendation model employing a graph attention network to model recursive dynamic social diffusion across the user and item spaces, thereby generating user and item representations.

3. LightGCN(He et al., 2020): A lightweight GCN employing linear propagation and aggregation among neighbors to compute user-item embedding vectors.

4. SocialLGN(Liao et al., 2022): A model integrating user-item interaction graphs with social graphs through a propagation mechanism based on lightweight GCN, facilitating its application in social recommendation.

5. MHCH(Yu et al., 2021b): This model introduces a multi-channel hypergraph convolutional network that enhances social recommendations by accounting for higher-order relationships and constructing multi-hop paths within heterogeneous information networks. MHCH can more profoundly comprehend the intricate associations between user interests and item features, thereby offering more accurate recommendations.

6. SEPT(Yu et al., 2021a): A socially-aware self-supervised triple training framework employing a neighbor identification contrastive learning approach, utilizing pseudo-labels from neighbors to refine user representations. Under the self-supervised triple training paradigm, it effectively harnesses social relationships to augment the performance of the recommendation model.

The above are all highly representative methods in the recommendation field. SBPR belongs to the traditional matrix factorization-based methods, using the user-item interaction matrix to learn latent vector representations of users and items. Diffnet++, LightGCN, and SocialLGN are all graph neural network-based methods that leverage graph structure to better capture the relationships between users and items. MHCH goes further by utilizing hypergraph structure and adopting self-supervised learning to improve recommendation effectiveness. SEPT is also based on graph neural networks and self-supervised learning, utilizing social information to generate views for contrastive learning. These methods all have high practical value and have driven the development of social recommendation systems.

D Model Parameter Settings

All experiments used the Adam optimizer with an initial learning rate uniformly set to 0.001. To further improve the convergence efficiency of the model, a learning rate decay strategy was adopted, which dynamically adjusts the learning rate during training to avoid the model getting stuck in local optima or oscillating. The initialization of model parameters used a Gaussian distribution with a mean of 0 and a standard deviation of 0.1 to ensure that the initial weights were randomly distributed within a reasonable range. To prevent overfitting, L2 regularization was introduced in the experiments, with the same regularization coefficient of 0.0001 applied to all baselines. Additionally, to maintain consistency across experiments, all baselines employed a two-layer graph convolutional structure and uniformly utilized 64-dimensional embedding vectors for experimentation.

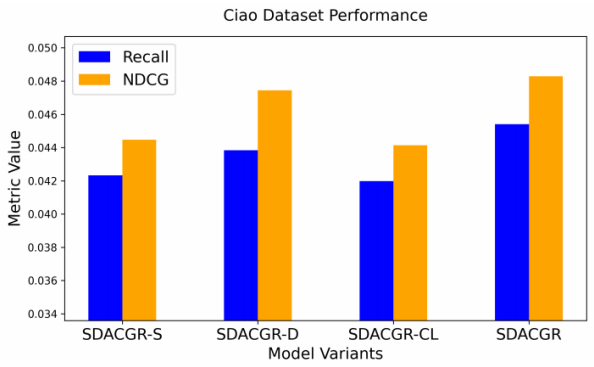
E Conclusion of baseline performance

SBPR, a conventional matrix factorization approach, fails to capture higher-order interactions between users and items, resulting in inferior performance compared to graph neural network-based models. Diffnet++, LightGCN, and SocialLGN, all graph neural network-based methods, possess enhanced modeling capabilities and can leverage higher-order information and structural features. By aggregating neighboring features, they can extract higher-order semantics, thereby surpassing matrix factorization methods. Moreover, graph adjacency information and higher-order propagation mechanisms facilitate the use of neighboring node information to enrich features, even in sparse datasets. Among these, SocialLGN further incorporates social information; however, due to the ambivalent impact of social information on recommendations, SocialLGN does not demonstrate a significant advantage over LightGCN, both having their respective strengths and weaknesses.

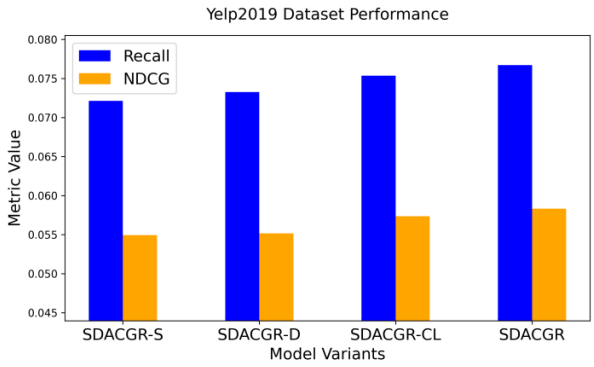
MHCH and SEPT exhibit the best performance among all baselines, with their key characteristic being the employment of self-supervised learning to enhance recommendation effectiveness. MHCH utilizes a hypergraph structure to capture higher-order information within social networks, modeling intricate multi-hop relationships between users and items to thoroughly investigate latent semantic associations. Additionally, MHCH effectively integrates information from different hypergraph structures through its self-supervised learning mechanism, enhancing the model's expressive capability and recommendation performance. SEPT, conversely, establishes a dual-task self-supervised enhancement framework predicated on the homogeneity prior of graph neural networks: on one hand, it identifies substructural patterns of user-item interactions via a sub-discriminative task to bolster local feature extraction; on the other hand, it harmonizes the global semantic space across views via a domain-discriminative task, thereby revealing latent supervisory signals within the behavioral and attribute perspectives of the original data.

F Dataset difference analysis

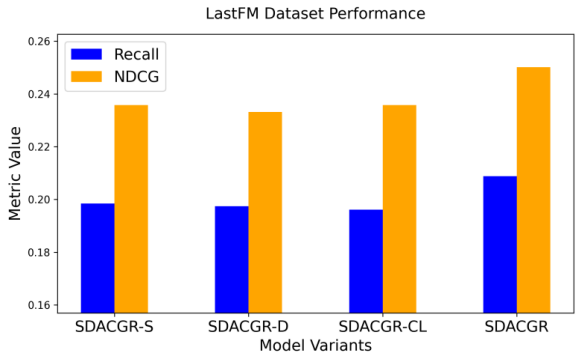
The experimental results are depicted in Fig. 4. Owing to variations in the datasets, including information density, the ratio of social interactions to total interactions, and the efficacy of social information, the roles of each module differ accordingly. In the Ciao dataset, the contrast learning module exerts the most significant impact, as this dataset has the lowest interaction density, necessitating self-supervised learning to address data sparsity issues; the Yelp dataset, wherein social information is nearly twice the amount of interaction information, exhibits a significant impact on recommendation performance upon the exclusion of social information; LastFM represents the most unique case, as the low credibility and high noise levels of user social information lead to diminished performance when utilizing unprocessed social information, as opposed to omitting social information entirely.



(a) Ciao dataset



(b) Yelp2019 dataset



(c) LastFM dataset

Figure 4: Ablation experiment results on three datasets

G Experiment figure

G.1 Impact of the Number of Contrastive Views K

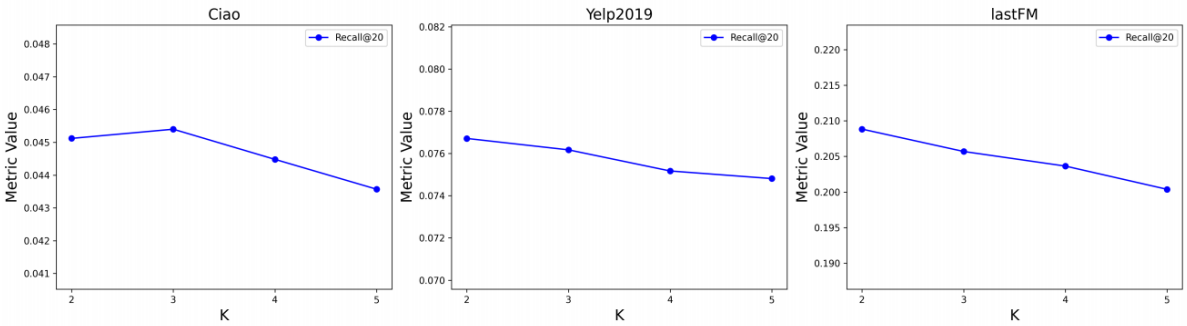
The adaptive view generation module employs multi-head GAT to generate K augmented views for contrastive learning. Since K directly affects both representation quality and computational cost, we evaluate different settings.

As shown in Fig. 5, the best performance on Yelp2019 and LastFM is achieved when $K = 2$. On Ciao, performance improves slightly when K increases from 2 to 3, but decreases as K becomes larger. Excessive augmented views introduce additional complexity and reduce training efficiency. Overall, $K = 2$ or $K = 3$ provides the best balance between effectiveness and efficiency.

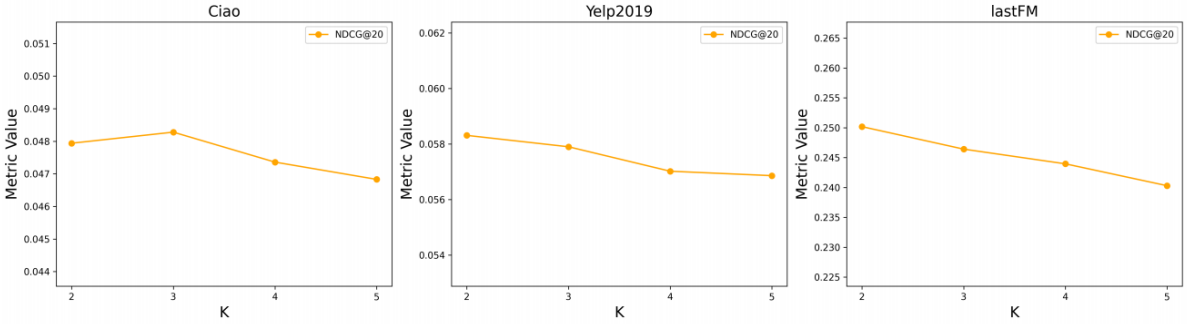
G.2 Impact of the Number of GNN Layers

The number of graph propagation layers L is another important hyperparameter. A shallow architecture may fail to capture higher-order dependencies, whereas a deeper architecture may suffer from overfitting.

To investigate its impact, we evaluate SDACGR with $L \in \{1, 2, 3, 4\}$ on the three datasets. As shown in Fig. 6, the optimal setting for Ciao, Yelp2019, and Gowalla is $L = 3$. When $L = 1$ or 2, higher-order structural information is insufficiently captured. Increasing the depth to $L = 4$ results in performance degradation, likely due to overfitting. These results suggest that incorporating social information benefits from moderate-depth propagation, which effectively captures higher-order dependencies while maintaining generalization.

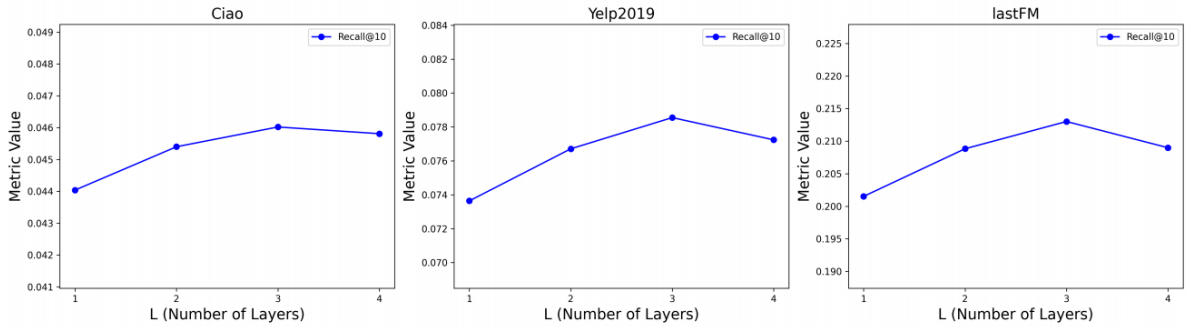


(a) Recall@10

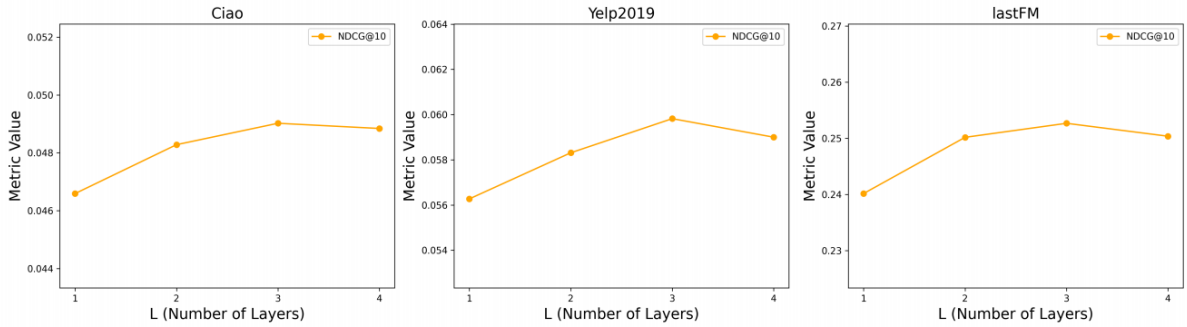


(b) NDCG@10

Figure 5: Number of comparison views K Experiment



(a) Recall@10



(b) NDCG@10

Figure 6: Experiment on the number of propagation layers in graph neural networks