# Disentangled and Diffusion-Denoising Enhanced GNN for Social Recommendation

Xinlong Yang, Dan Yang, Yang Liu

*Abstract*—**Social recommender systems face challenges in modeling complex user preferences due to underexplored item semantic relatedness, ignored connections are driven by complex factors, and noise may be introduced by social relation data. To address these limitations, we propose DDGNN4Rec (Disentangled and Diffusion-Denoising Graph Neural Networks for Social Recommendation), a novel framework that integrates disentangled representation learning with diffusion-denoising techniques. First, a collaborative heterogeneous graph is built, incorporating social connections between users, interactions between users and items, and relationships among items. Furthermore, we design novel memory enhancement mechanisms for message propagation and aggregation within the framework of graph neural networks, enabling the automatic and iterative distillation of semantic relationships into the representations of both users and items. Second, a latent-space diffusion-denoising module is introduced to improve the robustness of social relation data through multi-step forward noise propagation and reverse denoising training in the latent space. Experiments on three real-world datasets (Ciao, Epinions, Yelp) demonstrate that that DDGNN4Rec achieves advanced performance. Ablation studies validate the necessity of disentangled memory mechanisms, layer normalization, and diffusion-denoising.**

*Index Terms*—**Graph Neural Network; Social Recommendation; Memory-Enhanced Network; Diffusion-Denoising**

## I. INTRODUCTION

Traditional recommendation algorithms (e.g., collaborative filtering) rely on user-item interaction data but suffer from significant performance degradation in data-sparse or cold-start scenarios. Social recommendation mitigates these issues by incorporating user social relations, leveraging social network homophily to enhance user preference modeling. Early social recommendation models (e.g., SocialMF[1]) integrated social trust relations through matrix factorization but struggled to model complex high-order interactions. Recently, Graph Neural Networks (GNNs) have emerged as a research focus in social recommendation due to their powerful representation learning capabilities for graph-structured data. Through message passing mechanisms, GNNs aggregate neighborhood information from both user-item interaction graphs and social graphs, simultaneously capturing collaborative filtering signals (user-item interactions) and social homophily (user-user associations)[2]. For instance, GraphRec[3] first applied GNNs to social recommendation with a dual aggregation framework encoding user-item interactions and social influence separately; DiffNet[4] simulated dynamic social influence propagation through deep influence diffusion; DANSER[5] introduced dual attention mechanisms to distinguish explicit (e.g., direct following) and implicit (e.g., shared communities) social influences in user-user graphs, dynamically fusing multi-source information.

However, existing social recommendation models typically couple social relations with user behaviors into a single graph structure, leading to three limitations:

- The semantic correlations among projects are rich, but this is something that most existing learning solutions have yet to explore. In real life, items often exhibit inherent relations (e.g., product types/functionalities, spatial proximity of venues). Such semantic correlations help reveal latent dependencies crucial for understanding complex user interests.
- The driving force of connection is influenced by complex factors, but most current social recommendation systems ignore this point. User-item interactions (e.g., clicks/purchases) may be driven by diverse item attributes (brand/color for products, director for movies). Similarly, social connections form through multifaceted motivations (interest-based communities, colleagues, family). If no distinction is made when representing inter-user influence, it is difficult for the learned user preferences to reflect multiple social contexts.
- Social relations may contain irrelevant interactions (e.g., off-topic communications) contradicting preference similarity. While social data captures interest-aligned relations, noise from spurious connections could mislead recommendations—overemphasizing similarities between socially connected users with limited shared interests. GNN-based methods are particularly vulnerable as their message passing amplifies biases from noisy social edges.

To address these challenges, we propose a Disentangled and Diffusion-Denoising Graph Neural Networks for Social Recommendation (**DDGNN4Rec**), investigating social recommendation through disentangled learning of heterogeneous latent factors. We designed a memory network based on node type and edge type to solve the problem of relational heterogeneity and implement Disentangled relational modeling, which preserves independent feature representations for distinct interaction types (user-user, item-item, user-item). The algorithm makes use of external memory units that incorporate different embedding

Manuscript received April 2, 2025; revised July 19, 2025.

Xinlong Yang is a postgraduate student at School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (e-mail: 1367142724@qq.com.com).

Dan Yang is a professor at School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (corresponding author to provide e-mail: asyangdan@163. com).

Yang Liu is an associate professor at School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (e-mail: liuyang_lnas@163.com).

propagation operators. This approach allows graph architectures to directly model the different relationships between social data and user project data. Instead of separately parameterizing each relation type, our method encodes relation heterogeneity in disentangled latent spaces through memory neural network layers, operating in a fully automated and interactive manner without relying on handcrafted meta-paths. Second, we integrate the discriminative power of generative diffusion models with denoising training objectives, implementing refined noise diffusion and removal processes to help social recommender systems effectively handle connection noise. Finally, recommendation lists are predicted using learned node embeddings.

Our key contributions include:

- We propose DDGNN4Rec (Disentangled and Diffusion-Denoising Graph Neural Networks for Social Recommendation), a novel framework achieving generalized encoding of relation heterogeneity through relation-aware disentangled representations.
- We used an efficient latent space diffusion-denoising module. Through multi-step noise propagation and denoising training, the algorithm gains robust noise-handling capabilities to process diverse social connections and generate accurate user preference representations.
- Our method and baseline model were placed on the dataset for experiments to obtain the experimental results. Detailed ablation studies validate the framework's effectiveness and quantify each component's contribution to performance improvement.

## II. Related Work

This section introduces four categories of recommendation algorithms: social relation-based, graph neural network-based, disentangled-based, and context-aware recommendation methods.

### *A. Classic Social Recommendation Algorithms*

To enhance recommendation performance, numerous social recommendation approaches have been developed that incorporate online social relationships as supplementary data within recommendation systems [6]. Conventional techniques such as Sorec [7] and TrustMF [8] typically rely on matrix factorization, projecting users into a latent factor space. These methods are generally motivated by the assumption that users tend to exhibit similar item preferences to their friends within social networks [9]. Owing to the strong capability of neural networks in representing knowledge, social recommendation models based on deep learning [10] have attracted growing interest. In particular, several works utilize graph convolutional networks to jointly capture user–user and user–item interactions, as evidenced by DiffNet [4], RecoGCN [11], and KCGN [12]. Moreover, attention mechanisms have been incorporated to model varying degrees of social influence during preference learning, exemplified by SAMN [9] and GraphRec [3]. For instance, GraphRec accounts for differential strength in social relations when integrating information from both social and interaction graphs. Recent methods, inspired by self-supervised learning, have also adopted data augmentation strategies; examples include MHCN [14] and SMIN [15].

### *B. Graph Neural Network-Based Recommendation Algorithms*

Owing to the benefits of graph-structured data for representation learning, numerous studies in recommendation have concentrated on improving the modeling of user-item interactions by employing architectures based on graph neural networks (GNN) [16]. Building upon the success of spectral graph convolutional networks, NGCF [17] introduces an approach to capture high-order user-item relations through the use of convolutional operations. Another GNN-based recommendation research explores spatial GNNs for targeted aggregation of neighborhood information, such as KGAT[18] and DGRec[19]. Furthermore, heterogeneous graph representation has emerged as a promising solution in GNN research for integrating diverse relation contexts into node embeddings[20]. For example, HERec[21] incorporates auxiliary information through meta-path-based connections to enhance user preference learning. Encode different relationships by leveraging various types of user-item interactions (for example, clicks, purchases) [22]. Unlike these methods, the proposed DDGNN4Rec model automatically captures heterogeneous relations across users and items. Moreover, existing encoders for heterogeneous contexts have not explored latent factor encoding with disentangled representations.

### *C. Context-Aware Recommendation Algorithms*

Several studies have focused on developing context-aware recommendation algorithms[23] by incorporating various contextual signals in different scenarios. From the user perspective, user-user relations derived from online social networks are integrated with certain user and the user's context in STARS [24] to enhance collaborative filtering. LBSNs[25] improves point-of-interest recommendations by learning mappings between item contextual features and user preference tags. CARL[26] fuses textual context information of items with interaction-based embeddings for enhanced representation learning. Additionally, knowledge graphs are regarded as valuable contextual signals for integration into recommendation systems, boosting the performance of knowledge-aware methods such as KGIN[27], CASR[28] and KGCL[29].

### *D. Disentangled Recommendation Algorithms*

Recent studies have explored learning disentangled representations from user-item interactions[30]. For example, DGCF[31] designs a capsule network-based routing mechanism to model disentangled user-item relations. A curriculum learning-based approach is proposed in Work [32] to disentangle multi-type user feedback. In multimedia recommendation, weakly supervised disentangled representation learning is adopted for multi-modal features[33]. In contrast, the DDGNN4Rec method leverages different memory networks to capture latent factor-wise interdependencies across multiple disentangled representation spaces. Through this process, the relation-aware latent factors are thus capable of retaining distinct heterogeneous semantics.

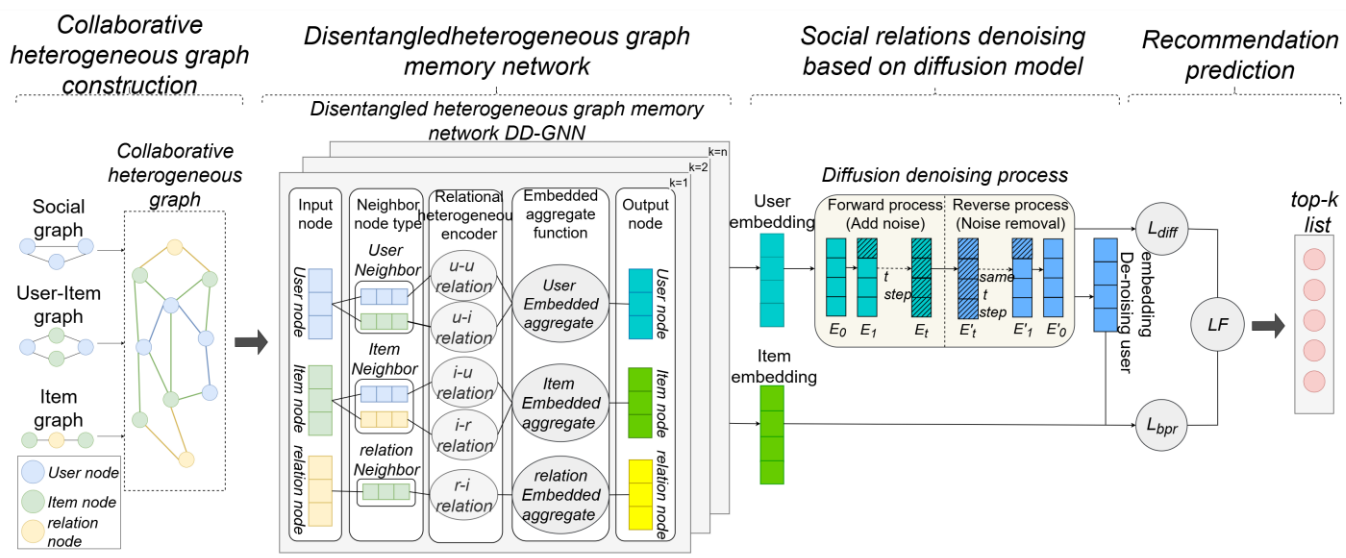


Fig. 1. Framework of DDGNN4REC

## III. SOCIAL RECOMMENDATION FRAMEWORK DDGNN4REC

This section proposes a disentangled Graph Neural Network to perform disentangled representation learning for heterogeneous relations in social recommendation. By introducing item-wise relations, we extend the data information utilized by the collaborative filtering framework. To achieve disentangled representation learning on heterogeneous relation data, we construct a memory-augmented network tailored to different node types and edge types, enabling distinct representation space decomposition for heterogeneous nodes and edges. A diffusion-denoising module is incorporated to enhance social recommendation performance by denoising social relations between users. The overall framework is illustrated in Figure 1.

### A. Collaborative Heterogeneous Graph Construction

To comprehensively integrate social connections, user-item interactions, and item-wise relations. we combine these three types of data to construct a collaborative heterogeneous graph. Item-wise relations data are organized according to relation types. This graph contains user nodes, item nodes, and relation nodes, where edges between users and items represent user-item interactions, edges between users denote social connections, and item-wise relations are expressed via "item-relation-item" triples.

We use ***Y*** to represent interactions. Similarly, we use ***S*** and ***T*** to represent social connections and item-wise relations. To store ***Y, S*** and ***T***, we define a graph structure $G=(D,E,I,J)$, using the nodes and edges in graph ***G*** to store ***Y, S*** and ***T***. We use ***I*** and ***J*** to represent the set of nodes and the set of edges. In Graph ***G***, the mapping functions of nodes and edges are $D \to I$ and $E \to J$ respectively. The formula for describing the different relationships represented by different connections between nodes in Graph G is as follows:

$$D=U\cup V\cup R;\quad E=S\cup T\cup Y \qquad (1)$$

In the formula, U, V, and R respectively represent users, items, and item relationship nodes.. Through this step, the heterogeneous graph is unified into three types of relationship nodes and the connections between nodes.

### B. Disentangled Memory Network

Most current methods for heterogeneous graph analysis rarely explicitly model the underlying factors that generate complex semantics. Leveraging the capability of memory neural networks in learning disentangled representations, we enhance graph neural networks with specialized memory modules that operate over multiple latent spaces to learn representations. By adaptively mapping node and edge types to latent factors, the disentangled heterogeneous semantics are more effectively retained.

Throughout message passing, we first apply localized feature transformation and nonlinear activation, then proceed with the integration of relation-aware contextual representations. Formally, this process can be described as (from layer l to l+1):

$$H^{(l+1)}[t] \leftarrow \underset{\forall s\in N(t)}{Aggre}\left(\varphi\left(H^{(l)}[t],H^{(l)}[s],e_{s,t}\right)\right) \qquad (2)$$

In the formula, $\boldsymbol{H^{(l)}[t]}$ denotes the latent representation of target node $t$ at the $l$-th layer, and $\boldsymbol{N(t)}$ represents the set of neighboring nodes (source nodes $s$) connected to $t$ via edges $e_{s,t}$.

#### 1) Relation Heterogeneity Encoder

For a collaborative graph with three heterogeneous relationships, we model the relationship heterogeneity by using specific embedded projections of vertex and edge types through external storage units. Each edge type (i.e. social relation, item relation and interaction relation) propagates information independently through node type and edge type dependent embedding projections. Specifically, we define ***M*** as a collection of memory units corresponding to latent factors. The prototype learning function $\boldsymbol{\phi(\cdot)}$ for relation-specific semantics is formulated as:

$$\varphi\left(H^{(l)}[t],H^{(l)}[s]\right)=\left(\sum_{m=1}^{|M|}\eta\left(H^{(L)}[t],m\right)W_{m}^{1}\right)H^{(l)}[s] \qquad (3)$$

$$\eta\left(H^{(l)}[t],m\right)=\sigma\left(H^{(l)}[t]\cdot W_m^2+b_m\right) \tag{4}$$

In the formula, $\eta(\cdot)$ refers to an embedding function specific to the target node. Trainable transformation matrices and biases are $W_m^1 \in \mathbb{R}^{d\times d}$, $W_m^2 \in \mathbb{R}^d$ ,and $b_m \in \mathbb{R}$ while $H^{(l)}[s] \in \mathbb{R}^d$ and $H^{(l)}[t] \in \mathbb{R}^d$ represent encoded feature embeddings of the source nodes and target nodes, respectively.

*2) Embedding Aggregation Function*

After encoding heterogeneous relation attributes from local neighbors, we aggregate propagated messages for each node:

**a. For user nodes**, we aggregate messages from social relations and interactions, the procedure can be formalized as:

$$H^{(l+1)}[u_i]=\frac{1}{|N_{u_i}^S|+|N_{u_i}^Y|}\left(\sum_{u_i'\in N_{u_i}^S}\varphi\left(H^{(l)}[u_i'],H^{(l)}[u_i]\right)+\sum_{v_i\in N_{u_i}^Y}\left(\sum_{m=1}^{|M|}\eta\left(H^{(l)}[v_j],m\right)W_{i\leftarrow j}^{m,1}\right)H^{(l)}[u_i]\right) \tag{5}$$

In the formula, $|N_{u_i}^S|$ and $|N_{u_i}^Y|$ are the number of neighbors in the social graph and interaction graph for user $u_i$, respectively. $W_{i\leftarrow j}^{m,1} \in \mathbb{R}^{d\times d}$ is a mapping matrix from the item representation space to the user space for the m-th disentangled latent factors.

**b. For item nodes,** we aggregate messages from item-wise relations and user-item interactions, the procedure can be formalized as:

$$H^{(l+1)}[v_i]=\rho_{i,j}\left(\sum_{v_j\in N_{v_j}^Y} Meg_{v_j\leftarrow u_i}^{(l)}+\sum_{r\in N_{v_j}^T} Meg_{v_j\leftarrow r}^{(l)}\right) \tag{6}$$

In the formula, $\boldsymbol{\rho_{i,j}}$ is a normalization term $\rho_{i,j}=1/|N_{v_j}^Y|+|N_{v_j}^T|$. Propagated messages ( $Meg_{v_j\leftarrow u_i}^{(l)}$, $Meg_{v_j\leftarrow r}^{(l)}$ ) are determined by the memory-based encoding function $\boldsymbol{\phi(\cdot)}$.

**c. For relation nodes** a similar approach is adopted, the procedure can be formalized as:

$$H^{(l+1)}[r]=\frac{1}{|N_r|}\sum_{v_j\in N_r}\left(\sum_{m=1}^{|M|}\eta\left(H^{(l)}[v_j],m\right)W_{r\leftarrow j}^{m,1}\right)H^{(l)}[r] \tag{7}$$

In the formula, $\boldsymbol{N_r}$ is the neighbor set of meta-relation node **r**, and $W_{r\leftarrow j}^{m,1} \in \mathbb{R}^{d\times d}$ is the m-th memory-specific transformation from the item space to the meta-relation node space.

*3) Self-Loop Propagation and Layer Normalization*

To stabilize training, DDGNN4Rec generalizes the heterogeneous graph message aggregation mechanism by integrating self-loop propagation and layer normalization, the formula is shown below:

$$\tilde{H}^{(l+1)}[v]=\sigma(\omega_1\odot\frac{H^{(l+1)}[v]-\mu}{\sqrt{\sigma^2+\varepsilon}}+\omega_2)+\varphi(H^{(l)}[v]) \tag{8}$$

In the formula, $\omega_1$ and $\omega_2$ are learnable scaling and bias terms, $\mu$ and $\sigma$ represent the mean and variance of input $H^{(l+1)}[v]$, and $\odot$ is element-wise multiplication. For self-loop connections, DDGNN4Rec applies the relation heterogeneity encoder $\boldsymbol{\phi(\cdot)}$ instead of directly adding embeddings from previous layers.

To leverage multi-layer node embeddings, we perform cross-layer (L-step) aggregation of higher-order embeddings:

$$H^*[v]=LN(\tilde{H}^{(0)}[v]\,||\,...\,||\,\tilde{H}^{(L)}[v]) \tag{9}$$

In the formula, $H^*[v] \in \mathbb{R}^d$ is the final embedding of node v, $\boldsymbol{LN(\cdot)}$ denotes layer normalization, and || represents vector concatenation..

## C. *Social Relation Denoising via Diffusion Models*

Inspired by the success of diffusion models in generating noise-free data across domains, we use a diffusion model to produce denoised social relation data. To address the inherent sparsity of social graph data, we implement an approach that enables efficient and effective social diffusion by conducting forward and reverse diffusion processes in the latent space, rather than the graph data space.

- **Forward Process:** Gaussian noise is incrementally added to the original social relation data E*s*, gradually transforming it into pure Gaussian noise.
- **Reverse Process:** A trainable neural network is leveraged to eliminate noise, generating refined social graph data.

The denoising process is formalized as $G_S \xrightarrow{\phi} E^S \xrightarrow{\psi} \bar{E}^S \xrightarrow{\psi'} \hat{E}^S \xrightarrow{\phi'} \hat{G}^S$ where $\phi$ and $\phi'$ denote bidirectional projection functions between the graph data domain (**G***s*) and the latent embedding domain (**E***s*).$\Psi$ and $\psi'$ represent the forward and reverse diffusion processes. Specifically, $\psi$ introduces noise to **G***s*, while $\psi'$ is optimized in the latent space to remove noise.

## D. *Top-N Recommendation Prediction*

To inject social influence into the prediction stage, DDGNN4Rec refines the learned user embeddings via a representation recalibration function $\tau(\cdot)$. The mathematical expression of $\tau(\cdot)$ is:

$$\tau(\mathbf{H}^*[ui])=\frac{1}{|N_{u_i}^S|+1}\left(\sum_{u_i'\in N_{u_i}^S}\mathbf{H}*[u_i']+\mathbf{H}^*[u_i]\right) \tag{10}$$

This function averages the embeddings of socially connected users, directly incorporating social information into subsequent predictions. Formally:

$$\xi(u_i,v_j)=(H^*[u_i]+\tau(H^*[u_i]))^T\cdot H^*[v_j] \tag{11}$$

**Optimization Objective**: The training objective is defined as a composite loss integrating pairwise BPR loss and weight decay regularization terms:

$$L=\sum_{(i,j^+,j^-)\notin O}-\log\delta(\xi(i,j^+)-\xi(i,j^-))+\lambda\sum_t L_t' \tag{12}$$

In the formula, $L_t'$ denotes the **diffusion loss**, and $t$ represents the sampled diffusion step for user $u$.

# IV. EXPERIMENTS AND EVALUATION

## A. *Dataset*

Three real-world benchmark datasets that are widely used in social recommendation were selected for the experiment: **Ciao**, **Epinions** and **Yelp**. Table I summarizes the statistical details of the three datasets.

Table I
Statistics of the Experimental Datasets

| Datasets | Number of users | Number of items | User-item interaction density | Number of interactions | Number of social connections | Social relation density |
|---|---|---|---|---|---|---|
| Ciao | 1925 | 15053 | 0.1048% | 30370 | 65084 | 1.7564% |
| Epinions | 18081 | 251722 | 0.0157% | 715821 | 572784 | 0.1752% |
| Yelp | 99262 | 105142 | 0.0074% | 769929 | 1298522 | 0.0132% |

- **Yelp:** This dataset originates from the Yelp platform, capturing users feedback on venues. It includes social relations between users, with a focus on social networks formed by individuals with shared interests.
- **Ciao:** The Ciao dataset is derived from the Ciao platform, which records user reviews and ratings for various products and services. It details social interaction information between users, particularly emphasizing social networks shaped by common preferences and interactive engagement.
- **Epinions:** The Epinions dataset is collected from the Epinions platform, a social network-based review system, and includes user evaluations of diverse products. Its unique feature lies in subdividing the 1-to-5 rating system into five explicit interaction categories: Negative, Below Average, Neutral, Above Average, and Positive.

### *B. Evaluation Metrics*

The experimental results focus on the Top-N recommendations and adopt two evaluation indicators that are widely used in the research of social recommendations: Hit Rate (HR) and Normalized Discounted Cumulative Gain (NDCG), both measuring recommendation accuracy based on Top-N ranked positions. For each target user during evaluation, 100 non-interacted items are selected as negative samples and combined with interacted items. The evaluation metrics are formalized as:

$$HR = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} r_{i,j}}{M} \tag{13}$$

$$NDCG = \sum_{i=1}^{M} \frac{\sum_{j=1}^{N} r_{i,j} / \log_2(j+1)}{M \cdot IDCG_i} \tag{14}$$

In the formula, M is the number of test users. For the i-th user, $r_{i,j} = 1$ if the j-th item in their ranked list is a positive sample; otherwise, $r_{i,j} = 0$. The numerator of NDCG@N is the Discounted Cumulative Gain (DCG)@N, while IDCG represents the maximum possible DCG@N value for the i-th test user.

### *C. Baselines*

To comprehensively evaluate the performance of this algorithm, we compares it with the latest methods from different research directions, including the following categories:

**Attentive Social Recommender Systems**: Attention mechanisms are effective techniques for identifying critical relations in social recommendations.

- **SAMN[9]:** Utilizes a two-stage attention network, incorporating a friend-wise attention mechanism, to capture social influence from relevant friends for enhanced user preference modeling..
- **EATNN[34]:** Employs attention mechanisms within a transfer learning framework to fuse interaction and social data, utilizing a multi-task optimization scheme.

**GNN-based Social Recommendation Models:** The social recommendation system that utilizes graph neural networks (GNNS) integrates user-item interaction data with user-user social connections. By propagating information in this unified graph structure, GNN learns the overall node representation to improve recommendation performance.

- **DiffNet[4]:** Uses a hierarchical diffusion architecture to model social relations. Simulates recursive social influence via dynamic social diffusion.
- **GraphRec[3]:** Propagates embeddings on social networks using graph attention networks. Aggregates social connections and item interactions into latent user representations via attention.
- **MHCN[14]:** A self-supervised multi-channel hypergraph neural network that captures user relations. Maximizes mutual information between node-level and subgraph-level embeddings as an auxiliary task.

**Graph Collaborative Filtering Models:** Graph collaborative filtering captures collaborative effects through user-item interaction graphs.

- **GCCF[35]:** A simplified GNN-based model using convolutional message passing. Removes non-linear transformations to mitigate overfitting.
- **NGCF[17]:** A graph convolution-based model that injects collaborative signals by modeling high-order connectivity with recursively applying graph propagation functions.

**Temporal-aware Social Recommendation:** Methods incorporating temporal contexts into social influence modeling.

- **DGRec[19]:** Integrates temporal dynamics using recurrent units and GNNs. Embeds social connections into dynamic user interest representations.

**Disentangled Graph Recommender Systems:** Compared with methods using disentangled learning techniques

- **DisenHAN[36]:** Encodes disentangled embeddings via graph attention, distinguishing user-item connections for propagation
- **DGCF[31]:** Partitions user embeddings into disentangled intents and performs intent-aware message passing on GNNs.

**Knowledge-aware Recommender System:** Leverages knowledge graphs (KGs) for item semantic relevance.

- **KGAT[18]:** A knowledge-enhanced model aggregating information from user-item interactions and KGs via attention.:

**Heterogeneous Graph Representation for Recommendation:**

- **HAN[37]:** Encodes heterogeneous graphs through a hierarchical attention network (node & semantic levels) guided by meta-paths.
- **HGT[37]:** A heterogeneous graph transformer with edge-specific attention and transformations.
- **HERec[38]:** A heterogeneous embedding method combining fusion functions and meta-path random walks.

TABLE II
PERFORMANCE COMPARISON OF ALL METHODS ON THE *HR@10* AND *NDCG@10* METRICS

| Datasets | Metrics | EATNN | SAMN | GraphRec | DiffNet | GCCF | NGCF | KGAT | DGRec | DisenHAN | DGCF | HGT | HAN | MHCN | HERec | DDGNN4Rec |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Ciao** | HR | 0.4130 | 0.4677 | 0.4594 | 0.5202 | 0.4926 | 0.4843 | 0.4907 | 0.5086 | 0.4856 | 0.5189 | 0.4933 | 0.4856 | 0.5080 | 0.5298 | **0.5591** |
| | Imp | 35.37% | 19.54% | 21.70% | 7.47% | 13.49% | 15.44% | 13.93% | 9.92% | 15.13% | 7.74% | 13.33% | 15.13% | 10.05% | 5.53% | – |
| | NDCG | 0.2520 | 0.2838 | 0.2670 | 0.3201 | 0.3070 | 0.3088 | 0.2977 | 0.3113 | 0.2894 | 0.3166 | 0.3062 | 0.2608 | 0.3118 | 0.3104 | **0.3493** |
| | Imp | 38.61% | 23.07% | 30.82% | 9.12% | 13.77% | 13.11% | 17.33% | 12.20% | 20.69% | 10.32% | 14.07% | 33.93% | 12.02% | 12.53% | – |
| **Epinions** | HR | 0.6422 | 0.6390 | 0.6865 | 0.6323 | 0.6779 | 0.6944 | 0.6756 | 0.6268 | 0.6825 | 0.6635 | 0.7001 | 0.6673 | 0.6411 | 0.6767 | **0.7425** |
| | Imp | 15.61% | 16.19% | 8.15% | 17.42% | 9.52% | 6.92% | 9.90% | 18.45% | 8.79% | 11.90% | 6.05% | 11.26% | 15.81% | 9.72% | – |
| | NDCG | 0.4483 | 0.4259 | 0.4786 | 0.4160 | 0.4783 | 0.4763 | 0.4708 | 0.4127 | 0.4627 | 0.4594 | 0.4812 | 0.4371 | 0.4261 | 0.4572 | **0.5272** |
| | Imp | 17.59% | 23.78% | 10.15% | 26.73% | 10.22% | 10.68% | 11.97% | 27.74% | 13.93% | 14.75% | 9.55% | 20.61% | 23.72% | 15.31% | – |
| **Yelp** | HR | 0.7273 | 0.7971 | 0.8019 | 0.8222 | 0.8130 | 0.8204 | 0.7737 | 0.7830 | 0.8159 | 0.7956 | 0.8185 | 0.8169 | 0.8019 | 0.7047 | **0.8381** |
| | Imp | 15.23% | 5.14% | 4.51% | 1.93% | 3.08% | 2.15% | 8.32% | 7.03% | 2.72% | 5.34% | 2.39% | 2.59% | 4.51% | 18.93% | – |
| | NDCG | 0.5289 | 0.5293 | 0.5372 | 0.5524 | 0.5585 | 0.5651 | 0.5386 | 0.5386 | 0.5403 | 0.5410 | 0.5547 | 0.5511 | 0.5348 | 0.4990 | **0.5895** |
| | Imp | 11.45% | 11.37% | 9.73% | 6.71% | 5.55% | 4.31% | 9.45% | 9.45% | 9.10% | 8.96% | 6.27% | 6.96% | 10.22% | 18.13% | – |

### D. *Parameter Setting*

We implements the proposed DDGNN4Rec based on the PyTorch framework, utilizing the Adam optimizer for model training. The key hyperparameter settings for DDGNN4Rec are as follows: the embedding dimension is tuned in the range of {4, 8, 16, 32}; the learning rate is fixed at 0.01; the batch size is adjusted in the range of [512 ,4096]; the regularization term coefficient λ is set in the range of $\{10^{-3}, 10^{-4}, 10^{-5}\}$; and the number of memory units is configured as 8 to balance the relation heterogeneity encoding capability and model complexity.

### E. *Experimental Results and Analysis*

Table II presents the experimental results of the selected baseline method and DDGNN4Rec on the three different selected datasets.

The following main findings can be summarized: Experiments on three datasets show that DDGNN4Rec achieves improvement on HR@10 and NDCG@10 metrics, and significantly outperforms the baseline models such as GraphRec and DiffNet. Compared with all baseline models, DDGNN4Rec achieves the best performance, which proves the superiority of DDGNN4Rec in performance. This performance improvement is attributed to the following model designs:

- DDGNN4Rec can retain comprehensive relation semantics while disentangling latent factors, thereby effectively integrating disentangled social and knowledge-aware collaborative signals.
- Through the integration of knowledge-aware item relations into the social recommendation framework, DDGNN4Rec improves the modeling of heterogeneous user-item relations, resulting in enhanced representational capacity for (user-item) interactions.
- Through multi-step noise propagation and removal training, DDGNN4Rec obtains strong denoising capabilities, effectively handling diverse social connections among users and generating accurate user preference representations.

The performance gap between DDGNN4Rec and other GNN methods indicates that neglecting item relations restricts the performance of social recommendation systems, while validating DDGNN4Rec 's capability to fully exploit latent factors in user and item domains through its graph neural structures that enhance memory. As observed in Table III, DDGNN4Rec outperforms other methods in performance improvements across different Top-N ranking positions, further verifying its superior ranking capability. Additionally, the recommendation accuracy improves as the N value increases.

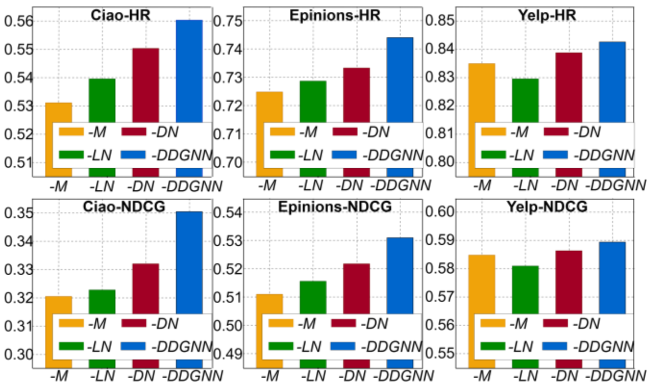

Fig. 2. Ablation Studies of different submodules

### F. *Model Ablation Study*

In this section, we examined the design rationality of the modules in the DDGNN4Rec framework. To this end, three model variants corresponding to DDGNN4Rec's three core technical components are constructed by removing key modules:

- "-M": DDGNN4Rec excluding the disentangled memory-enhanced relation heterogeneity encoder.
- "-LN": DDGNN4Rec that removes layer normalization in propagation layers to validate its role in stabilizing node embedding training;
- "-DN": DDGNN4Rec without the generative diffusion-denoising module.

TABLE III
PERFORMANCE EVALUATION OF *HR@N* AND *NDCG@N* ACROSS DIFFERENT TOP-N VALUES

| Datasets | Metrics | SAMN | EATNN | GraphRec | DiffNet | GCCF | NGCF | DGRec | DGCF |
|---|---|---|---|---|---|---|---|---|---|
| Ciao | HR@5 | 0.3465 | 0.2972 | 0.3059 | 0.3943 | 0.3683 | 0.3571 | 0.3725 | 0.3876 |
| | NDCG@5 | 0.2463 | 0.2122 | 0.2234 | 0.2813 | 0.2666 | 0.2362 | 0.2646 | 0.2783 |
| | HR@20 | 0.7497 | 0.7504 | 0.8006 | 0.7369 | 0.7908 | 0.8013 | 0.73 | 0.7777 |
| | NDCG@20 | 0.4558 | 0.4558 | 0.5014 | 0.4476 | 0.4854 | 0.5008 | 0.4422 | 0.4814 |
| Epinions | HR@5 | 0.5175 | 0.5284 | 0.5681 | 0.5104 | 0.5537 | 0.5613 | 0.5054 | 0.5475 |
| | NDCG@5 | 0.3863 | 0.3921 | 0.4323 | 0.3822 | 0.4163 | 0.4315 | 0.3774 | 0.4143 |
| | HR@20 | 0.7493 | 0.7509 | 0.8006 | 0.7364 | 0.7907 | 0.8015 | 0.7306 | 0.7778 |
| | NDCG@20 | 0.4554 | 0.4553 | 0.5016 | 0.4474 | 0.4853 | 0.5008 | 0.4424 | 0.4814 |
| Yelp | HR@5 | 0.6358 | 0.6426 | 0.6632 | 0.6703 | 0.6704 | 0.6744 | 0.6513 | 0.6563 |
| | NDCG@5 | 0.4667 | 0.4863 | 0.4901 | 0.5126 | 0.5135 | 0.5194 | 0.4891 | 0.4953 |
| | HR@20 | 0.9007 | 0.8068 | 0.8945 | 0.9056 | 0.9013 | 0.9012 | 0.8825 | 0.9012 |
| | NDCG@20 | 0.5409 | 0.546 | 0.5653 | 0.5702 | 0.5504 | 0.5687 | 0.5618 | 0.5674 |

| Datasets | Metrics | KGAT | HAN | DisenHAN | HERec | HGT | MHCN | **DDGNN4Rec** |
|---|---|---|---|---|---|---|---|---|
| Ciao | HR@5 | 0.3393 | 0.2939 | 0.3495 | 0.3833 | 0.3416 | 0.3868 | **0.4171** |
| | NDCG@5 | 0.2421 | 0.1896 | 0.2489 | 0.2678 | 0.2373 | 0.2796 | **0.2982** |
| | HR@20 | 0.7884 | 0.7762 | 0.7899 | 0.7798 | 0.8057 | 0.7494 | **0.8359** |
| | NDCG@20 | 0.4832 | 0.4807 | 0.4914 | 0.4833 | 0.5026 | 0.4558 | **0.5426** |
| Epinions | HR@5 | 0.5486 | 0.5402 | 0.5608 | 0.5517 | 0.5758 | 0.5198 | **0.6255** |
| | NDCG@5 | 0.4136 | 0.4104 | 0.4248 | 0.4178 | 0.4362 | 0.3884 | **0.4856** |
| | HR@20 | 0.7884 | 0.7765 | 0.7896 | 0.7793 | 0.8051 | 0.7494 | **0.8359** |
| | NDCG@20 | 0.4831 | 0.4803 | 0.4911 | 0.4838 | 0.5025 | 0.4555 | **0.5426** |
| Yelp | HR@5 | 0.6502 | 0.6634 | 0.6512 | 0.5831 | 0.6884 | 0.6606 | **0.7052** |
| | NDCG@5 | 0.49 | 0.5088 | 0.4941 | 0.4503 | 0.5132 | 0.4914 | **0.5378** |
| | HR@20 | 0.8796 | 0.8974 | 0.9048 | 0.8127 | 0.9069 | 0.8956 | **0.9293** |
| | NDCG@20 | 0.5526 | 0.5527 | 0.5654 | 0.5036 | 0.5806 | 0.5673 | **0.6043** |

These ablation studies aim to verify the effectiveness of each module in capturing user social influence, handling relation heterogeneity, and stabilizing the training process.

The performance of the above variants and the full DDGNN4Rec is evaluated on the three datasets, as shown in Figure 2. DDGNN4Rec consistently outperforms all variants. By analyzing the results, the following conclusions are drawn:

- Impact of the Memory Mechanism: The removal of the memory-enhanced relation heterogeneity encoder (the “-M” variant) results in a considerable drop in performance, which confirms the efficacy of disentangling latent factors across heterogeneous relation types..
- Role of Layer Normalization: Comparing the "-LN" variant with DDGNN4Rec reveals that layer normalization critically contributes to training stability. This improvement is attributed to stable gradients induced by normalization, mitigating fluctuations in node embeddings.
- Denoising Module Necessity: Removing the generative diffusion-denoising module (i.e., the "-DN" variant) causes notable performance decline, confirming the necessity of denoising social relation embeddings.

These findings underscore the necessity of DDGNN4Rec’s core designs: the memory mechanism (disentangling heterogeneous relations), layer normalization, and diffusion-based denoising jointly ensure robustness and expressive power in complex interaction scenarios.

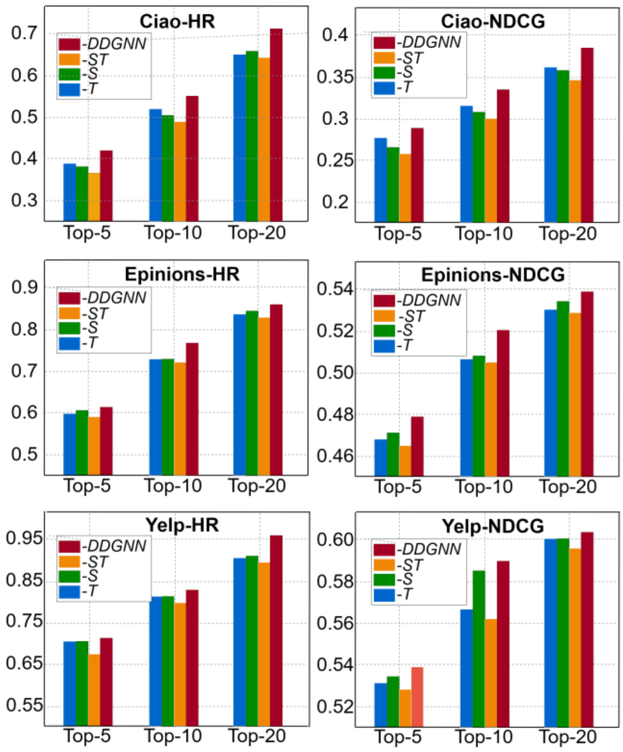


Fig. 3. Ablation studies on different input.

## G. *Effect of Heterogeneous Relationships*

In this section, we further investigate the influence of

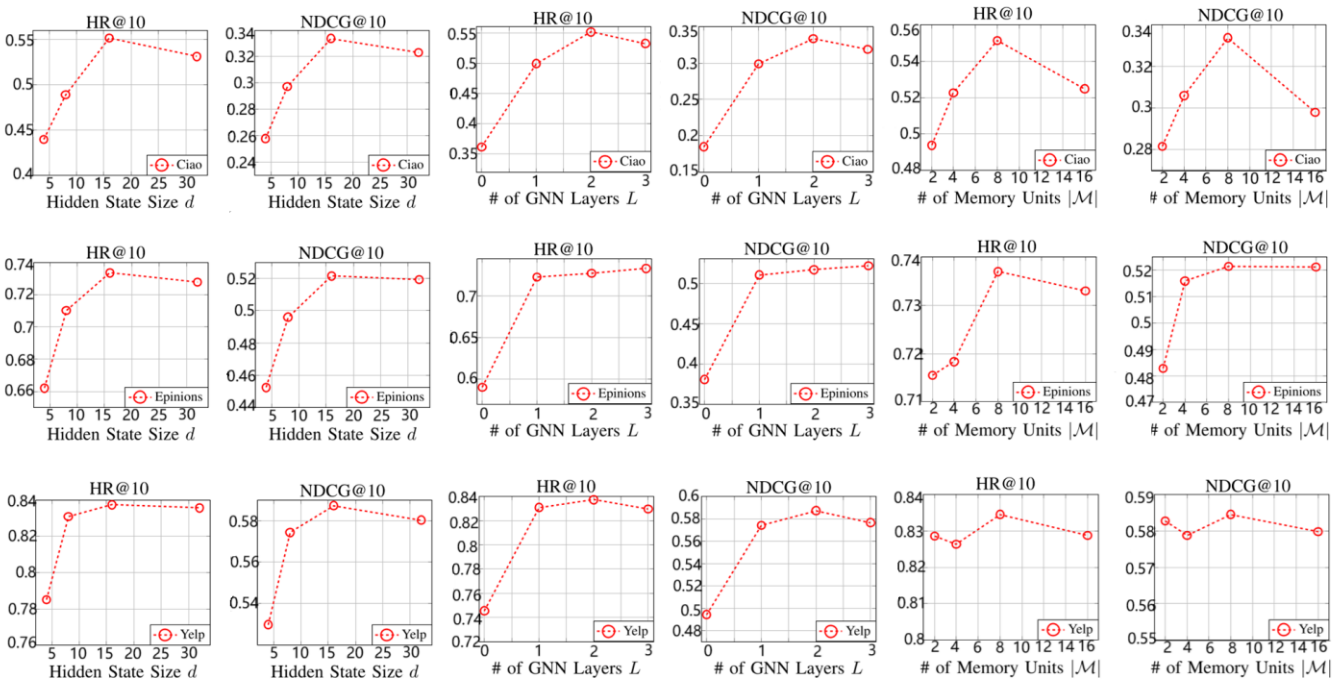


Fig 4. Hyperparameter studies of DDGNN4Rec

different auxiliary relational data on the performance of our DDGNN4Rec. To this end, the following three model variants are considered:

- "-T": DDGNN4Rec removes the item relation matrix $T \in \mathbb{R}^{J\times|R|}$;
- "-S": DDGNN4Rec without the user-user social relation matrix $S \in \mathbb{R}^{I\times I}$;
- "-ST": DDGNN4Rec only uses the user-item relationship as input.

The performance is evaluated under varying top-N settings. The outcomes are presented in Figure 3, from which the following key findings can be observed:

- The inclusion of auxiliary heterogeneous relations consistently contributes positively to model performance, owing to the integration of different contents into the representations.
- The relatively lower performance of the “-S” variant underscores the value of incorporating social contextual signals to enhance user preference learning.
- DDGNN4Rec consistently outperforms “-T” across all scenarios. We attribute this gain to the model's ability to capture rich itemized dependencies through memory-enhanced relational heterogeneous encoders.
- The “-ST” variant consistently produced the worst results on both datasets, proving different data from users or projects is helpful for the accuracy of the model..

### H. *Parameter Analysis*

This section studies the sensitivity of hyperparameter Settings by exploring the impact of each parameter used in DDGNN4Rec on model performance. Experimental results are shown in Figure 4.

The experiments reveal the following:

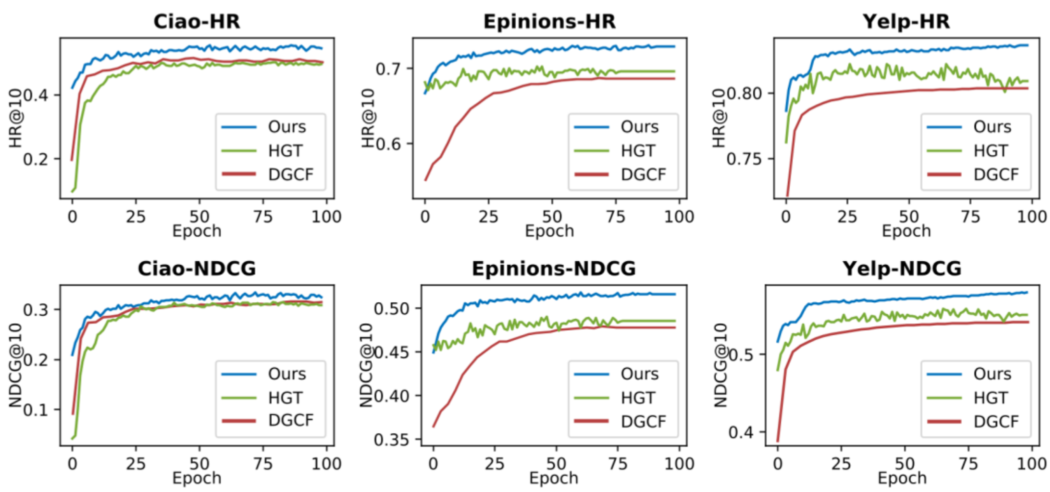


Fig. 5. Model Efficiency Study of DDGNN4Rec

- Hidden State Dimension d: Tested with values in the range of {4,8,16,32} (i.e., {$2^2$, $2^3$, $2^4$, $2^5$}), DDGNN4Rec achieves peak performance at d=16 due to its fusion of user-item dual-domain semantic correlations. Increasing d to 32 leads to performance degradation from parameter redundancy, confirming that smaller dimensions ensure model efficiency and practicality for real-world recommendation systems.
- Number of GNN Layers L: For $1 \le L \le 3$, two-hop neighborhood message passing (L=2) significantly improves performance by capturing high-order user-item relations. However, stacking to three layers (L=3) causes minor performance decline due to over-smoothing, while compare with the non-propagation variant (L=0) eliminates critical relation modeling capability.
- Number of Memory Units M: Tested in the range of {2,4,8,16} (i.e., {$2^1$, $2^2$, $2^3$, $2^4$}), setting M=8 achieves optimal heterogeneous relation encoding by disentangling latent factors. Excessive units (e.g., 16) introduce noise, while insufficient units (e.g., 2) limit semantic capture capacity.

Overall, DDGNN4Rec achieves the best balance between model efficiency, relation representation power, and training stability under the configuration d=16, L=2, and M=8.

### *I. Model Efficiency Study*

This section assesses the model efficiency by monitoring the convergence speed throughout the training rounds. An experimental comparison is conducted between two baselines, namely DGCF and HGT. Comparison of experimental data reflecting the progressive improvement achieved through parameter optimization. The results illustrated in Figure 5 lead to the following findings: Compared with HGT and DGCF, the experimental data of DDGNN4Rec performs better.. This underscores the efficacy of our model's parameter inference optimization. The architecture of DDGNN4Rec attains higher recommendation accuracy and demonstrates greater optimization ease. We attribute this advantage to the mapping of edge relations into multiple latent spaces, which effectively incorporates relation heterogeneity..

## V. CONCLUSIONS

We introduce a Disentangled and Diffusion-Denoising Graph Neural Networks for Social Recommendation (DDGNN4Rec). In order to handle relational heterogeneity and complete relational modeling of disentangled, a memory-enhanced network dependent on node and edge types is designed, which preserves distinct feature representations for various interaction types. Concurrently, the discriminative power of generative diffusion models is incorporated with a denoising training objective, employing an optimized noise diffusion and elimination procedure to allow social recommender systems to manage varied connection noises efficiently. Ultimately, recommendation lists are generated using the obtained node embeddings. Comprehensive assessments on three public datasets show that DDGNN4Rec yields competitive results against state-of-the-art approaches.

## REFERENCES


[1] H. Ma et al., "SoRec: Social Recommendation Using Probabilistic Matrix Factorization," in CIKM, 2008.
[2] W. Hamilton et al., "Representation Learning on Graphs: Methods and Applications," in IEEE Data Eng. Bull., 2017.
[3] W. Fan et al., "Graph Neural Networks for Social Recommendation," in WWW, 2019.
[4] L. Wu et al., "DiffNet++: A Neural Influence and Interest Diffusion Network for Social Recommendation," in TKDE, 2020.
[5] Q. Wu et al., "DANSER: Dual Graph Attention Networks for Deep Latent Representation of Multifaceted Social Effects in Recommender Systems," in IJCAI, 2019.
[6] Jin B , Cheng K , Zhang L ,et al.Partial Relationship Aware Influence Diffusion via a Multi-channel Encoding Scheme for Social Recommendation[J].ACM, 2020.
[7] Ma H, Yang H X, Lyu M R, et al. SoRec: Social recommendation using probabilistic matrix factorization[C]//Proceedings of the 17th ACM Conference on Information and Knowledge Management, New York：ACM，2008.
[8] B. Yang, Y. Lei, J. Liu, and W. Li. Social collaborative filtering by trust. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 39(8):1633–1647, 2016.
[9] Tang J , Hu X , Liu H .Social recommendation: a review[J].Social Network Analysis and Mining, 2013, 3(4):1113-1133.
[10] Fu B , Zhang W , Hu G ,et al. Dual Side Deep Context-aware Modulation for Social Recommendation[J]. 2021.
[11] Xu F , Lian J , Han Z ,et al. Relation-Aware Graph Convolutional Networks for Agent-Initiated Social E-Commerce Recommendation[C]//Conference on Information and Knowledge Management. ACM, 2019.
[12] Huang C , Xu H , Xu Y ,et al. Knowledge-aware Coupled Graph Neural Network for Social Recommendation[J]. 2021.
[13] Chen C , Zhang M , Liu Y ,et al. Social Attentional Memory Network: Modeling Aspect- and Friend-Level Differences in Recommendation[J].ACM, 2019.
[14] Yu J , Yin H , Li J ,et al. Self-Supervised Multi-Channel Hypergraph Convolutional Network for Social Recommendation[J]. 2021.
[15] Long X , Huang C , Xu Y ,et al. Social Recommendation with Self-Supervised Metagraph Informax Network[J]. 2021.
[16] Chen L , Wu L , Hong R ,et al. Revisiting Graph based Collaborative Filtering: A Linear Residual Graph Convolutional Network Approach[J].Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(1):27-34.
[17] Wang X , He X , Wang M ,et al. Neural Graph Collaborative Filtering[C]//the 42nd International ACM SIGIR Conference. ACM, 2019.
[18] Wang X , He X , Cao Y ,et al. KGAT: Knowledge Graph Attention Network for Recommendation[J].ACM, 2019.
[19] Song W , Xiao Z , Wang Y ,et al. Session-Based Social Recommendation via Dynamic Graph Attention Networks[J].ACM, 2019.
[20] M. Chen, C. Huang, L. Xia, W. Wei, Y. Xu, and R. Luo. Heterogeneous graph contrastive learning for recommendation. In International Conference on Web Search and Data Mining (WSDM), pages 544–552,2023.
[21] Shi C , Hu B , Zhao X ,et al. Heterogeneous Information Network Embedding for Recommendation[J].IEEE Transactions on Knowledge & Data Engineering, 2017:1-1.
[22] T. Gu, C. Wang, C. Wu, Y. Lou, J. Xu, C. Wang, K. Xu, C. Ye, and Y. Song. Hybridgnn: Learning hybrid representation for recommendation in multiplex heterogeneous networks. In International Conference on Data Engineering (ICDE), pages 1355–1367. IEEE, 2022.
[23] Adomavicius G , Bauman K , Tuzhilin A ,et al.Context-Aware Recommender Systems: From Foundations to Recent Developments[J]. 2022.
[24] Kulkarni S , Rodd S F .Context Aware Recommendation Systems: A review of the state of the art techniques[J].Computer science review, 2020, 37(Auga):100255.1-100255.33.
[25] Aliannejadi M , Crestani F .Personalized Context-Aware Point of Interest Recommendation[J].Acm Transactions on Information Systems, 2018, 36(4):1-28.
[26] Wu L , Quan C , Li C ,et al. A Context-Aware User-Item Representation Learning for Item Recommendation[J]. 2017.
[27] Wang X , Huang T , Wang D ,et al. Learning Intents behind Interactions with Knowledge Graph for Recommendation[J]. 2021.
[28] Mezni H , Benslimane D , Bellatreche L .Context-Aware Service Recommendation Based on Knowledge Graph Embedding[J].IEEE Transactions on Automatic Control, 2022, 34(11):14.

[29] Chen G , Xie X .ML-KGCL: Multi-level Knowledge Graph Contrastive Learning for Recommendation[C]//International Conference on Database Systems for Advanced Applications. Springer, Cham, 2023.
[30] Qian T , Liang Y , Peng S Z .Intent Disentanglement and Feature Self-Supervision for Novel Recommendation[J].IEEE Transactions on Knowledge and Data Engineering, 2023, 35(10):9864-9877.
[31] Wang X , Jin H , Zhang A ,et al. Disentangled Graph Collaborative Filtering[J].ACM, 2020.
[32] Chen H , Chen Y , Wang X ,et al. Curriculum Disentangled Recommendation with Noisy Multi-feedback[C]//Neural Information Processing Systems.2021.
[33] Wang X , Chen H , Zhu W .Multimodal Disentangled Representation for Recommendation[J].IEEE, 2021.
[34] Chen C , Zhang M , Wang C ,et al. An Efficient Adaptive Transfer Neural Network for Social-aware Recommendation[C]//the 42nd International ACM SIGIR Conference. ACM, 2019.
[35] Chen L , Wu L , Hong R ,et al. Revisiting Graph based Collaborative Filtering: A Linear Residual Graph Convolutional Network Approach[J].Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(1):27-34.
[36] Liu M , Chua T S , Wang X ,et al. KGAT: Knowledge Graph Attention Network for Recommendation.2019[2025-09-01].
[37] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu. Heterogeneous graph attention network. In The Web conference (WWW), pages 2022–2032. ACM, 2019.
[38] Shi C , Hu B , Zhao X ,et al. Heterogeneous Information Network Embedding for Recommendation[J].IEEE Transactions on Knowledge & Data Engineering, 2017:1-1.