

CGMR-DSR: Cross-graph multi-relational dynamic social recommendation

Weihua Xu*, Yibo Li

College of Artificial Intelligence, Southwest University, Chongqing, 400715, China

ARTICLE INFO

Communicated by L. Oneto

Keywords:

Social recommendation
Cross-graph interaction
Multi-relational modeling
Graph neural network

ABSTRACT

Classical recommendation approaches, including matrix factorization and conventional graph-based social recommendation models, mainly rely on historical user-item interactions or a single social-link view. As a result, they often fail to jointly exploit interaction and social graphs, combine direct social links with behavior-derived similarity links, and capture the temporal evolution of user preferences. To address these limitations, this paper proposes CGMR-DSR, a cross-graph multi-relational dynamic social recommendation model. The main contributions are threefold. First, CGMR-DSR models two complementary social relation types and employs basis decomposition-enhanced R-GCN to learn relation-aware social representations with controlled parameter complexity. Second, it performs direct bidirectional feature exchange between shared user nodes in the interaction and social graphs, followed by user-item graph propagation that transfers the social-enhanced user signals to items. Third, it integrates LSTM-based temporal modeling and adaptive alignment loss to capture dynamic preference evolution and suppress noisy social influence. Experiments on seven benchmark datasets evaluate the competitiveness of CGMR-DSR against representative and recent baselines on HR@10 and NDCG@10, together with ablation analyses of its major components.

1. Introduction

Over the past decade, the deep integration of social platforms and e-commerce has driven the rapid development of social recommendation technology [1]. This integration has not only reshaped the ways users access information and make consumption decisions, but has also transformed the interaction patterns between platforms and users [2]. With the continuous growth in the scale of social users and the explosive accumulation of user behavior data, the problem of information overload has become increasingly prominent [3]. Users struggle to filter out information that matches their preferences from massive amounts of content [4]. As a core tool for solving this problem, recommendation systems are widely applied in social network scenarios [5]. By incorporating users' social relationship data, recommendation systems can better match content to user interests and may improve user engagement and platform outcomes. Consequently, social recommendation has become an active research area in both industry and academia [6,7].

Specifically, the fundamental premise of social recommendation systems lies in the fact that users' preferences and behaviors are notably affected by their social ties, including friends, followed entities, or trust-based relationships [8]. For example, a user may develop an interest in a certain product due to recommendations from social friends, or adjust

their consumption choices based on evaluations from trusted entities [9]. Recommendation models based on social relationships can provide content aligned with users' interests and help identify potential preferences. Social recommendation therefore seeks to learn user preferences from both interaction records and social connections [10]. A broad range of methods has been developed under this paradigm, including models that combine collaborative signals with social influence.

However, existing social recommendation systems still face three limitations in modeling temporal preferences and complex social relations:

- Fragmented multi-graph integration.* Most existing methods (e.g., DiffNet and LDGSR) jointly model user-item interaction graphs and social graphs but treat them as separate semantic spaces, leaving cross-graph node correspondence insufficiently exploited.
- Single-view social relationship modeling.* Many methods use only observed social links and do not distinguish them from behavior-derived user similarity.
- Limited temporal robustness.* Static frameworks cannot capture the temporal evolution of preferences and social influence, while sequential methods often omit heterogeneous social semantics and robustness to noisy links.

* Corresponding author.

Email addresses: chxuwh@swu.edu.cn (W. Xu), lybswu@163.com (Y. Li).

With the deep integration of social platforms and recommendation services, user preferences are shaped by several complementary signals. Historical behaviors reflect individual interests, observed social neighbors support preference diffusion, and training interactions reveal behavior-based similarity that may differ from explicit social links. Temporal factors can further affect whether earlier preferences remain informative. Existing studies usually focus on only one aspect of this problem, such as social influence propagation, graph structure aggregation, self-supervised denoising, or sequential modeling. The key research question in this study is how to jointly model these two social relation sources, interaction-social graph complementarity, dynamic historical preference, as well as robustness to noisy links within a trainable neural recommendation framework. CGMR-DSR addresses this question with relation-aware R-GCN encoding, topological matrix-based cross-graph propagation, LSTM-based dynamic modeling, and joint robust optimization.

The methodological scope of CGMR-DSR differs from heterogeneous knowledge-enhanced and cross-domain recommendation. It does not introduce external knowledge entities, define source and target domains, or transfer preferences across platforms. Instead, it couples two graph views from the same recommendation scenario through their shared user nodes: social-aware features are transferred to interaction-graph users, interaction-aware features are transferred back to social-graph users, and the social-enhanced user features subsequently update item representations through observed user-item interactions. The contribution is therefore a within-scenario graph-coupling mechanism that jointly models social relation semantics, interaction structure, and temporal preference evolution.

To tackle the aforementioned problems, this paper proposes CGMR-DSR, a cross-graph multi-relational dynamic social recommendation approach. The model combines basis decomposition-enhanced R-GCN with topological matrix-based user-level feature exchange to model two social relation types and interaction-social complementarity. The transferred social features first enhance user representations in the interaction graph and subsequently influence item representations through user-item graph propagation. LSTM-based temporal modeling captures the evolution of user preferences when a reliable event order is available, while the joint objective combines ranking, cross-graph robustness, and adaptive alignment losses to reduce the influence of noisy social links. Experiments on seven benchmark datasets evaluate the effectiveness of these components under a unified recommendation protocol.

The main contributions of this paper are summarized as follows:

1. A cross-graph multi-relational dynamic framework: CGMR-DSR uses topological matrices to exchange features directly between the shared user nodes of the multi-relational social graph and the user-item interaction graph. The resulting social-enhanced user features are then propagated to item nodes over observed user-item interactions, reducing information isolation between the two graph views.
2. Integration of two-relation social modeling and dynamic learning: By combining basis decomposition-enhanced R-GCN and LSTM, the model distinguishes direct social links from training-derived behavioral similarity while modeling sequential preference changes.
3. Reproducible evaluation protocol: The seven-dataset evaluation specifies dataset provenance, training-only graph construction, validation-based model selection, fixed candidate sets, HR@10, NDCG@10, and component ablations. Published and publicly reported results from 17 representative and recent baselines are retained as contextual references.

The remainder of this manuscript is organized as follows. [Section 2](#) reviews related studies on graph neural network-based social recommendation, self-supervised and denoising recommendations, and sequential or dynamic recommendations. [Section 3](#) presents the proposed

CGMR-DSR framework, including the problem formulation, multi-relational social graph modeling, cross-graph fusion, dynamic user embedding learning, and objective function. [Section 4](#) reports the experimental settings, datasets, baseline methods, evaluation metrics, performance comparison, ablation studies, and hyperparameter sensitivity analysis. [Section 5](#) concludes the paper and discusses future research directions.

2. Related work

2.1. Graph neural networks in social recommendation

Graph Neural Networks (GNNs) have become a core technique for social recommendation because they can represent user-item interactions and user-user relations as graph structures and learn higher-order collaborative signals through neighborhood aggregation. Early graph-based methods mainly focused on item-item or user-item relations. For example, ItemRank constructs an item graph and uses random walks to estimate item relevance, but it does not explicitly model social influence. Later GNN-based social recommendation methods, such as GraphRec, DiffNet, DGRec, NGCF+, and LDGSR, improve recommendation performance by incorporating social neighbors, higher-order connectivity, dynamic attention, or disentangled user representations into neural graph learning [2,14–17].

Recent studies have further shifted attention from simple social information aggregation to robust and scalable social graph learning. GBSR introduces an information bottleneck view for social recommendation and learns compact social structures by filtering redundant or noisy relations [11]. SGIL employs graph invariant learning to obtain stable user preference representations under noisy social environments [18]. Burger combines graph denoising-augmentation fusion with multi-semantic modeling to improve robustness in social recommendation [19]. PULSE, a 2026 graph-based social recommendation method, studies socially-aware user representation learning from a parameter-efficient perspective and highlights the scalability challenge in graph collaborative filtering with social knowledge [20]. These studies demonstrate that recent social recommendation research increasingly emphasizes noise resistance, stable representation learning, and scalability. However, most of them still focus on refining one social graph or one interaction graph, while the bidirectional interaction between user-item graphs and multi-relational social graphs remains insufficiently explored.

2.2. Self-supervised and denoising learning for recommendation

Self-supervised learning provides an effective way to alleviate data sparsity and noise interference in recommendation by constructing auxiliary supervision signals from unlabeled data. In graph collaborative filtering, contrastive learning is commonly used to generate augmented graph views and enforce representation consistency across different perturbations. Representative methods such as SGL and MHCN learn robust user and item embeddings by contrasting node representations under different graph augmentations or hypergraph channels [21,22]. In social recommendation, DSL aligns denoised embeddings from social and interaction views to reduce the negative influence of noisy social relations, while ESGL improves social recommendation by combining scale adjustment and self-supervised learning strategies [10,23].

More recent work has attempted to make self-supervised recommendation more adaptive and noise-aware. AdasRec introduces adaptive self-supervised learning for social recommendation and assigns different weights to multiple auxiliary tasks according to their contribution [24]. This line of research suggests that auxiliary objectives should not be treated as fixed regularizers, because improperly weighted self-supervision may interfere with the main recommendation task. Nevertheless, most existing self-supervised social recommendation methods mainly align representations between predefined views or perform graph perturbation on a single graph. They generally do not jointly

model direct social links and behavior-derived similarity links, cross-graph message transfer, and temporal preference evolution within a unified objective.

2.3. Sequential and dynamic recommendation

Sequential and dynamic recommendation methods aim to capture the temporal evolution of user preferences from historical interaction sequences. Traditional sequential recommendation models use recurrent neural networks, attention mechanisms, or session-based structures to encode recent behaviors. With the development of GNNs, session-based and sequential graph recommendation methods have been proposed to model transitions among interacted items and dynamic user interests. DGRec introduces dynamic graph attention for session-based social recommendation, showing the value of combining social influence and temporal behaviors [16]. Streaming and session-based GNN methods further demonstrate that temporal neighborhoods can improve the modeling of evolving user intentions [4].

Recent sequential graph methods also integrate self-supervised learning to improve dynamic representation learning. SelfGNN constructs short-term collaborative graphs and long-term interest representations for sequential recommendation, providing a recent example of combining graph neural networks with sequential preference modeling [25]. However, these methods usually focus on temporal user-item interactions and do not explicitly distinguish direct social links from behavior-derived similarity links. In contrast, CGMR-DSR jointly captures these two social relation types: direct user-level exchanges between interaction and social graphs, indirect social enhancement of item representations, and time-aware preference evolution. This distinguishes the proposed framework from existing GNN-based, self-supervised, and sequential recommendation models.

2.4. Heterogeneous knowledge, cross-domain, and continual recommendation

Heterogeneous knowledge-enhanced recommendation incorporates auxiliary entities, attributes, or knowledge graphs to enrich preference representations. For example, Wu et al. [34] combine historical conversational preferences with knowledge-graph information to improve personalized conversational recommendation. These methods focus on exploiting semantic knowledge or auxiliary entity relations, whereas CGMR-DSR focuses on the structural complementarity between a user-item interaction graph and a multi-relational social graph. It does not introduce an external knowledge graph or transfer semantic knowledge from auxiliary entities.

Cross-domain recommendation transfers preference knowledge from one or more source domains to a target domain, commonly requiring domain-specific and domain-invariant representations, overlapping entities, or an explicit transfer objective. Zhao et al. [33], for instance, study domain-oriented knowledge transfer to improve target-domain recommendation. CGMR-DSR is different: both graphs are constructed from the same recommendation scenario and share the same user set. The matrix C aligns user identities across graph views rather than aligning users or items across domains, and the model neither defines source and target domains nor transfers preference knowledge between platforms. Thus, its cross-graph fusion should be understood as within-scenario multi-view graph integration, not cross-domain recommendation.

Continual and adaptive recommendation further considers how a model incorporates newly arriving data while retaining knowledge acquired from previous stages. Ma et al. [35] formulate all-domain continual pre-training for recommendation by adapting a model across behavioral corpora from multiple domains. In contrast, the LSTM and time-decay modules in CGMR-DSR model the order and recency of interactions within a fixed training period; they do not perform continual parameter updates, replay, or explicit forgetting mitigation. Extending

CGMR-DSR with continual updates for newly arriving interactions and social links is therefore a promising direction for future work.

3. Proposed method: CGMR-DSR

3.1. Preliminaries of social recommendation

Let $U = \{u_1, \dots, u_N\}$ and $T = \{t_1, \dots, t_M\}$ denote the user and item sets. Each observed interaction is represented as a tuple $(u, i, \tau) \in D$, where $u \in U$, $i \in T$, and τ is a source-provided chronological position; a numerical timestamp is available only when it is supplied by the source. The user-item interaction graph is defined as $G_{ut} = (V_{ut}, E_{ut})$, where $V_{ut} = U \cup T$ and $E_{ut} = \{(u, i) \mid (u, i, \tau) \in D\}$. Its adjacency matrix is denoted by $A_{ut} \in \{0, 1\}^{(N+M) \times (N+M)}$.

The multi-relational social graph is defined as $G_{uu} = (U, E_{uu}, R)$, where R is the set of social relation types and $E_{uu} \subseteq U \times U \times R$. A triple $(u, v, r) \in E_{uu}$ indicates that users u and v are connected by relation r . The corresponding relation-specific adjacency matrices are denoted by $\{A_{uu}^{(r)}\}_{r \in R}$. For each user u , let $S_u = [(i_1, q_1), \dots, (i_{|S_u|}, q_{|S_u|})]$ denote the source-provided chronological interaction sequence extracted from the training set.

Given G_{ut} , G_{uu} , and the training sequences $\{S_u\}_{u \in U}$, the task is to learn a scoring function $f_\theta(u, i)$ that ranks unobserved candidate items for each user. At inference, the personalized list L_u contains the top- K items in $T \setminus T_u^+$ according to $f_\theta(u, i)$, where T_u^+ is the set of items previously interacted with by u . The subsequent sections define the relation encoder, user-level cross-graph exchange, temporal sequence encoder, and joint training objective used to parameterize f_θ .

3.2. Framework overview

Fig. 1 summarizes the CGMR-DSR framework. The design of CGMR-DSR follows a four-stage pipeline: relation-semantic encoding, cross-graph complementary fusion, temporal dynamic updating, and joint robust optimization. First, the user-item interaction graph is used to learn interaction-aware preference signals, while the multi-relational social graph is used to distinguish direct social links from behavior-derived similarity links. Second, the topological matrices C and C^T enable direct bidirectional feature exchange between the shared user nodes of the two graphs. The resulting social-enhanced user features are propagated over user-item interactions to update item representations indirectly. Third, source-provided interaction sequences are processed through LSTM and temporal attention; time decay is applied only when verified event-time fields are available. Finally, BPR loss, cross-graph robustness loss, and adaptive alignment loss are jointly optimized to improve ranking accuracy, cross-graph consistency, and robustness to noisy social links. This pipeline makes CGMR-DSR different from methods that only focus on single-graph propagation, static social relation modeling, or isolated sequential modeling.

3.3. Problem formulation and notation

Building on the definitions in the preliminaries, each user and item is initialized with trainable embeddings $x_u^{(0)} \in \mathbb{R}^{d_0}$ and $x_i^{(0)} \in \mathbb{R}^{d_0}$, respectively. The multi-relational encoder assigns a relation-specific transformation to every edge $(u, v, r) \in E_{uu}$, while the interaction encoder aggregates information over E_{ut} . The model learns a dynamic user representation $h_u^{\text{dyn}} \in \mathbb{R}^d$ and a stable item representation $h_i \in \mathbb{R}^d$, and predicts the preference score as $\hat{y}_{ui} = f_\theta(h_u^{\text{dyn}}, h_i)$. The ranking objective is defined in the Preference Prediction and Objective Function subsection.

For clarity, Table 1 summarizes the main notation used in the proposed framework. In this paper, we use u to denote a user, i to denote an item to avoid ambiguity, and τ to denote a temporal position in a user's historical sequence.

CGMR-DSR: Cross-Graph Fusion and Temporal Modeling for Recommendation

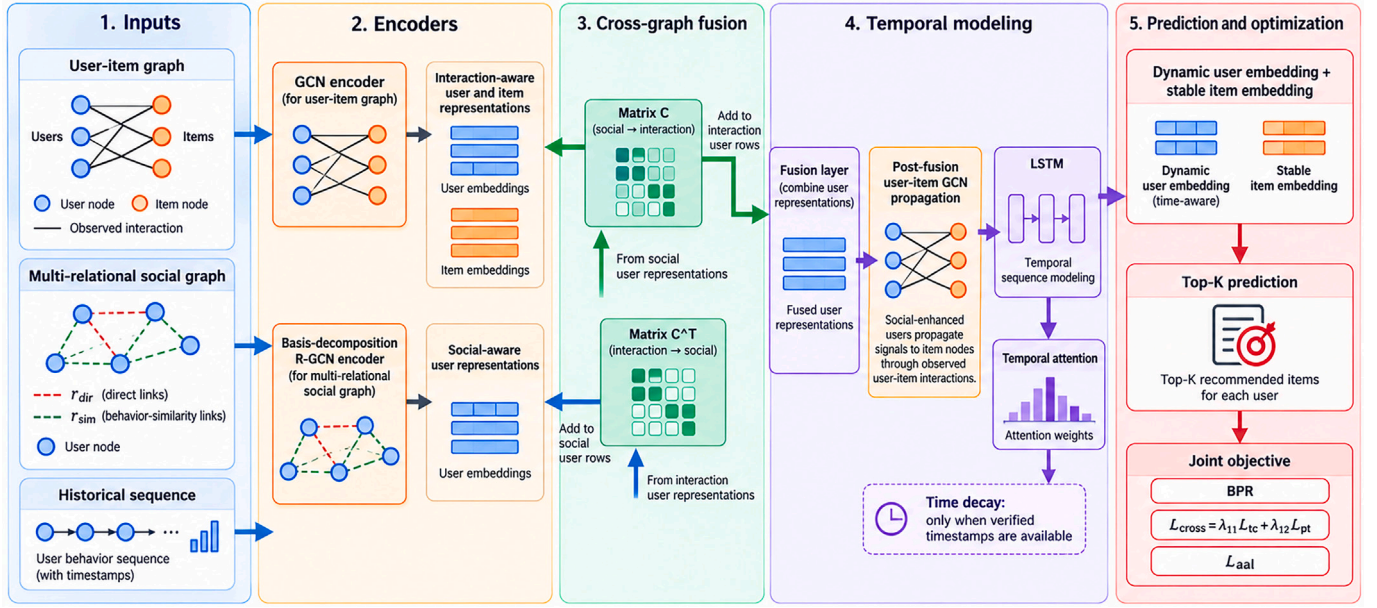


Fig. 1. Overview of the CGMR-DSR framework. The model first encodes the user-item interaction graph and the two-relation social graph through GCN and basis decomposition-enhanced R-GCN, respectively. The topological matrices C and C^T exchange features directly between their shared user nodes. Social-enhanced user representations are then propagated over the user-item graph, allowing items to receive social signals indirectly through observed interactions. Source-provided historical interactions are modeled by an LSTM with temporal attention; time decay is used only when verified event-time fields are available. Finally, dynamic user embeddings and stable item embeddings are used for top- K prediction, and the whole framework is optimized by the joint objective $\mathcal{L}_{total} = \mathcal{L}_{BPR} + \lambda_1 \mathcal{L}_{cross} + \lambda_2 \mathcal{L}_{aal}$.

Table 1
Summary of key notations.

Notation	Description
U, T	User set and item set
G_{ui}, G_{uu}	User-item interaction graph and multi-relational social graph
R	Set of social relation types
$A_{ui}, A_{uu}^{(r)}$	Adjacency matrix of the interaction graph and relation-specific adjacency matrices of the social graph
C	Cross-graph connection matrix between user nodes in G_{ui} and G_{uu}
H_{ui}^{int}	Interaction-aware node representations learned from G_{ui}
H_{uu}^{social}	Social-aware user representations learned from G_{uu}
$H_{ui}^{social}, H_{uu}^{int}, H_{uu}^{fused}$	Cross-graph transferred social features, interaction features, and fused social-graph user features
h_u^{dyn}, h_i	Dynamic representation of user u and stable representation of item i
S_u	Source-provided chronological interaction sequence of user u
R_{train}, Y	Training-only binary user-item matrix and fixed user pseudo-category labels
K_c	Number of pseudo categories used by MiniBatch K-means
$\mathcal{L}_{BPR}, \mathcal{L}_{cross}, \mathcal{L}_{tc}, \mathcal{L}_{pt}, \mathcal{L}_{aal}$	Ranking loss, cross-graph robustness loss, pseudo-category consistency loss, potential relationship loss, and adaptive alignment loss

3.4. Multi-relational social graph modeling with basis decomposition

To address the limitation of single-dimensional social relationship representation in traditional models, we construct a two-relation social graph and encode it using an R-GCN with basis decomposition [12]. The first relation type, r_{dir} , contains the observed trust or friendship links

supplied by each benchmark. The second type, r_{sim} , captures behavioral similarity derived only from training interactions. Specifically, for each user, we compute cosine similarity between L2-normalized rows of R_{train} , retain the ten most similar users with positive similarity, and symmetrize the resulting edges. Thus, $R = \{r_{dir}, r_{sim}\}$ and $E_{uu} = E_{dir} \cup E_{sim}$. This construction is performed once before model training, uses no validation or test interactions, and makes the relation semantics consistent across all seven datasets. Basis decomposition expresses the weight matrix of each relation type as a linear combination of shared basis matrices, reducing parameter complexity while allowing relation-specific transformations.

Formally, for each relationship type $r_k \in R$, the transformation matrix $W^{(r_k)}$ in the R-GCN layer is defined as:

$$W^{(r_k)} = \sum_{b=1}^B a_{k,b} W^{(b)} \quad (1)$$

where B denotes the validation-selected number of basis matrices, $a_{k,b} \in \mathbb{R}$ is the combination coefficient for relation r_k and basis b , and $W^{(b)} \in \mathbb{R}^{d \times d}$ denotes the b -th shared basis matrix. The embedding of user node v_i is updated by aggregating its neighbors across relation types:

$$h_i^{(l+1)} = \sigma \left(\sum_{r_k \in R} \sum_{v_j \in \mathcal{N}_i^{(r_k)}} \frac{1}{|\mathcal{N}_i^{(r_k)}|} W^{(r_k)} h_j^{(l)} + W^{(self)} h_i^{(l)} \right) \quad (2)$$

Here, l represents the layer of the R-GCN, $\mathcal{N}_i^{(r_k)}$ denotes the set of neighboring nodes of v_i under relationship r_k , $|\mathcal{N}_i^{(r_k)}|$ refers to the size of the neighbor set (employed for normalization), $W^{(self)}$ is the self-loop weight matrix, and $\sigma(\cdot)$ denotes the ReLU activation function. Following multiple layers of R-GCN propagation, we derive the social-aware user embedding $h_u^{social} \in \mathbb{R}^d$ for each user u .

3.5. Cross-graph fusion via topological matrix-based message passing

To reduce information isolation between the user-item interaction graph G_{ui} and the multi-relational social graph G_{uu} , this study proposes a topological matrix-based cross-graph message passing mechanism. The direct cross-graph exchange is defined on the user nodes shared by the two graphs: social-aware user features are transferred to the interaction graph, and interaction-aware user features are transferred back to the social graph. Item nodes do not have counterparts in the social graph and therefore do not receive a direct social feature transfer.

First, we build the adjacency matrix of the interaction graph $A_{ui} \in \mathbb{R}^{(N+M) \times (N+M)}$ and relation-specific adjacency matrices $\{A_{ui}^{(r)}\}_{r \in R}$ for the social graph. If there exists an interaction edge between node i and node j in G_{ui} , then $A_{ui}[i, j] = 1$; otherwise, it is zero. For each $r \in R$, $A_{ui}^{(r)}[i, j] = 1$ if the user pair (i, j) has relation r , and zero otherwise. The relation-specific matrices are consumed separately by the R-GCN.

Next, we define a cross-graph connection matrix $C \in \mathbb{R}^{(N+M) \times N}$ that maps the shared user nodes between the two graphs. For example, with two users ($U = \{u_1, u_2\}$) and three items ($T = \{t_1, t_2, t_3\}$), let $V_{ui} = \{u_1, u_2, t_1, t_2, t_3\}$, with user nodes listed first, and let $V_{uu} = \{u_1, u_2\}$. The connection matrix is

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The general rule is: $C[i, j] = 1$ if the i -th node in G_{ui} is the same user as the j -th node in G_{uu} ; otherwise, $C[i, j] = 0$. This matrix acts as a bridge for direct user-level feature transfer: social-aware user embeddings from G_{uu} are mapped to the user rows of G_{ui} through C , and interaction-aware user features from G_{ui} are fed back to G_{uu} through the transpose of C (C^T).

The cross-graph feature update formulas are as follows:

$$H_{ui}^{social} = C \cdot H_{uu}^{social} \quad (3)$$

$$H_{uu}^{int} = C^T \cdot H_{ui}^{int} \quad (4)$$

where $H_{ui}^{social} \in \mathbb{R}^{N \times d}$ denotes the user embedding matrix derived from the social graph (incorporating multi-relational weighted information), $H_{ui}^{social} \in \mathbb{R}^{(N+M) \times d}$ is the social feature matrix mapped onto the interaction graph, $H_{ui}^{int} \in \mathbb{R}^{(N+M) \times d}$ represents the feature matrix learned from the interaction graph, and $H_{uu}^{int} \in \mathbb{R}^{N \times d}$ is the interaction feature matrix fed back to the social graph. Because the item rows of C are zero, the item rows of H_{ui}^{social} are initially zero. These zero rows are not treated as final item-side social representations.

We fuse the interaction-aware and transferred social features for the user rows of the interaction graph, and fuse the social-aware and transferred interaction features for the user rows of the social graph:

$$\tilde{h}_u^{ui} = \text{Concat}(h_u^{int}, h_u^{social \rightarrow ui}) W_{\text{fusion}}, \quad \tilde{h}_i^{ui} = h_i^{int}. \quad (5a)$$

$$H_{uu}^{fused} = \text{Concat}(H_{uu}^{social}, H_{uu}^{int}) W_{\text{fusion}}^{uu}. \quad (5b)$$

The augmented user representations then propagate over the user-item interaction graph once more:

$$H_{ui}^{out} = \text{GCN}(\tilde{H}_{ui}, A_{ui}), \quad H_T^{fused} = \text{Slice}(H_{ui}^{out}, T), \quad (5c)$$

where $\tilde{H}_{ui}$ stacks the fused user rows $\tilde{h}_u^{ui}$ and the original item rows $\tilde{h}_i^{ui}$. Thus, items benefit indirectly from social information: social features first enhance their neighboring users and are subsequently propagated to item nodes through observed user-item interactions. In the reverse direction, interaction-aware user features are transferred from G_{ui} to G_{uu} through C^T and fused with social-aware user features in H_{uu}^{fused} . Therefore, the term “bidirectional” refers to direct user-level exchange between the two graphs, not to a direct item-to-social-graph mapping.

3.6. Dynamic user embedding learning with LSTM

Considering the temporal dynamics of user preferences, this study introduces an LSTM network [13] to model the sequential nature of users' historical interactions, thereby generating time-aware dynamic user embeddings. To robustly capture the evolutionary patterns of user preferences, we further optimize the model to address practical challenges such as inconsistent sequence lengths across users and differential impacts of interactions at varying time intervals.

CGMR-DSR explicitly handles sequential and historical information when the source provides a reliable interaction order. For each user, historical interaction records are arranged using source-provided chronological fields. Each record is represented by the interacted item and, where available, its timestamp and the interval from the previous interaction. Recent interactions can indicate short-term intent, whereas earlier interactions provide long-term preference signals. Therefore, the temporal module combines LSTM-based sequence modeling, mask-based valid-position filtering, and temporal attention aggregation; time-decay weighting is enabled only for records with verified event-time fields.

Formally, for a given user u , let $S_u = [(t_{u,1}, q_{u,1}), (t_{u,2}, q_{u,2}), \dots, (t_{u,T_u}, q_{u,T_u})]$ denote the chronological interaction sequence supplied by the data source, where $t_{u,\tau}$ denotes the interacted item, $q_{u,\tau}$ denotes the available ordering field, and T_u is the sequence length. When the source supplies numerical event times, $q_{u,\tau} = t_{s_{u,\tau}}$; otherwise, $q_{u,\tau}$ is an ordering key without an inferred interval interpretation. The corresponding item embedding sequence is $h_{S_u} = [h_{t_{u,1}}, h_{t_{u,2}}, \dots, h_{t_{u,T_u}}]$, where $h_{t_{u,\tau}} \in \mathbb{R}^d$ is the fused embedding vector of item $t_{u,\tau}$. Ties in verified timestamps are resolved by the source record order. We do not manufacture timestamps or time intervals from an arbitrary file order.

To handle inconsistent sequence lengths, we adopt a truncation-padding strategy combined with a mask mechanism: sequences shorter than a predefined maximum length $L_{\max}$ (determined by the 95th percentile of sequence lengths in the dataset) are padded with zero vectors at the end, while sequences longer than $L_{\max}$ are truncated from the earliest interactions to retain only the most recent $L_{\max}$ records. A binary mask vector $m_u \in \{0, 1\}^{L_{\max}}$ is introduced to mark valid interactions (1) and padded positions (0), ensuring padded values are ignored during model computation. This strategy enables the model to process users with different history lengths in the same mini-batch while avoiding artificially padded interactions from affecting the LSTM states or attention weights. Retaining the most recent records for long sequences is consistent with the assumption that recent interactions are more informative for short-term preference prediction.

For records with verified numerical event times, we incorporate the influence of intervals between consecutive interactions through time decay. The interval between two adjacent records is calculated as $\Delta\tau_\tau = t_{s_{u,\tau}} - t_{s_{u,\tau-1}}$ for $\tau \geq 2$, with $\Delta\tau_1 = 0$ for the first interaction. Larger intervals can indicate weaker temporal continuity. We map $\Delta\tau_\tau$ to a time-decay weight using an exponential function:

$$w_\tau = \exp\left(-\alpha \cdot \frac{\Delta\tau_\tau}{\max_{\tau'} \Delta\tau_{\tau'}}\right) \quad (6)$$

where $\alpha > 0$ is a hyperparameter controlling the decay rate (optimized via grid search) and $\max_{\tau'} \Delta\tau_{\tau'}$ normalizes time intervals to the range $[0, 1]$. This normalization is used only for genuine event times, regardless of whether their units are days or seconds. A larger value of w_τ indicates greater temporal proximity to previous behaviors. For a source without verified event-time fields, we set $w_\tau = 1$ and do not make time-interval or time-window claims for that source. The enhanced item embedding is obtained by fusing the original embedding with the time-decay weight:

$$\hat{h}_{t_{u,\tau}} = h_{t_{u,\tau}} \odot w_\tau \quad (7)$$

where $\odot$ denotes element-wise multiplication.

The normalized and enhanced sequence $\hat{h}_{S_u}^{\text{norm}}$ is fed into the LSTM, which processes the sequence step-by-step. The state update at each time step τ is as follows:

$$i_\tau = \sigma \left(W_i \cdot \text{Concat}(\hat{h}_{t_{u,\tau}}, h_{\tau-1}) + b_i \right) \quad (8)$$

$$f_\tau = \sigma \left(W_f \cdot \text{Concat}(\hat{h}_{t_{u,\tau}}, h_{\tau-1}) + b_f \right) \quad (9)$$

$$c_\tau = f_\tau \odot c_{\tau-1} + i_\tau \odot \tanh \left(W_c \cdot \text{Concat}(\hat{h}_{t_{u,\tau}}, h_{\tau-1}) + b_c \right) \quad (10)$$

$$o_\tau = \sigma \left(W_o \cdot \text{Concat}(\hat{h}_{t_{u,\tau}}, h_{\tau-1}) + b_o \right) \quad (11)$$

$$h_\tau = o_\tau \odot \tanh(c_\tau) \quad (12)$$

After the LSTM processes the sequence, the hidden state at each step contains historical interaction information and, where event times are verified, temporal-transition information. Instead of relying only on the final hidden state, we use temporal attention to assign different importance weights to valid historical states. The mask vector ensures that padded positions do not contribute to the attention denominator or the final dynamic user embedding.

To emphasize informative interactions, we integrate a temporal attention layer to aggregate the hidden states of all time steps. The attention weight for each time step is computed as:

$$a_\tau = \frac{\exp(\text{MLP}_{\text{attn}}(h_\tau) \cdot v_{\text{attn}}) \cdot m_u[\tau]}{\sum_{\tau'=1}^{L_{\text{max}}} \exp(\text{MLP}_{\text{attn}}(h_{\tau'}) \cdot v_{\text{attn}}) \cdot m_u[\tau']} \quad (13)$$

where $\text{MLP}_{\text{attn}}(\cdot)$ maps the hidden state to a query space, and v_{attn} is a learnable attention vector. The final dynamic embedding of user u is obtained by a weighted aggregation of the hidden states:

$$h_u^{\text{dyn}} = \sum_{\tau=1}^{L_{\text{max}}} a_\tau \cdot h_\tau \quad (14)$$

For items, since their attribute features are relatively stable compared to user preferences, the fused embeddings after cross-graph aggregation are used as the final stable embeddings, i.e., $h_i = h_i^{\text{fused}}$.

3.7. Preference prediction and objective function

Based on the dynamic user representation and the stable item representation, CGMR-DSR predicts the preference score between user u and item i using an inner product. To avoid ambiguity, i is used as the item index in this subsection, while τ is used for temporal positions. The prediction function is defined as:

$$\hat{y}_{ui} = h_u^{\text{dyn}} \cdot h_i^T \quad (15)$$

where h_u^{dyn} denotes the dynamic embedding of user u obtained from the LSTM and the temporal attention module, h_i denotes the stable embedding of item i after cross-graph feature fusion, and $\hat{y}_{ui}$ is the predicted preference score.

For pairwise ranking optimization, we construct training triples $D = \{(u, i, j) \mid i \in T_u^+, j \in T_u^-\}$, where T_u^+ is the set of items interacted with by user u and T_u^- is the set of sampled negative items. The BPR loss encourages the score of a positive item i to be higher than that of a negative item j :

$$\mathcal{L}_{\text{BPR}} = -\frac{1}{|D|} \sum_{(u,i,j) \in D} \ln(\sigma(\hat{y}_{ui} - \hat{y}_{uj})) \quad (16)$$

where $\sigma(\cdot)$ is the sigmoid function.

To enhance the robustness of cross-graph representation learning, we further introduce a cross-graph robustness loss. It consists of a pseudo-category consistency loss L_{tc} and a potential relationship loss L_{pt} :

$$\mathcal{L}_{\text{cross}} = \lambda_{11} L_{\text{tc}} + \lambda_{12} L_{\text{pt}} \quad (17)$$

Because none of the seven benchmark datasets provide unified user-category annotations, L_{tc} uses fixed pseudo-category labels derived from

training interactions only. Specifically, we construct a binary user-item matrix R_{train} , where $R_{ui} = 1$ if $(u, i) \in D_{\text{train}}$ and $R_{ui} = 0$ otherwise. Each row is L2-normalized and projected to a d_p -dimensional representation using Truncated SVD. MiniBatch K-means is then applied to these representations, and the cluster assignment of user u is used as its pseudo-category label $y_u \in Y$. The number of pseudo categories K_c is selected from $\{8, 16, 32\}$ according to validation performance, and the selected value for each dataset is reported in the experimental settings. This offline procedure is conducted before model training and does not use validation or test interactions.

Let h_u^{int} denote the user row of $H_{\text{ui}}^{\text{int}}$ and h_u^{soc} denote the user row of $H_{\text{uu}}^{\text{fused}}$. Two category prediction heads map these branch-specific representations to a probability distribution over pseudo categories. The pseudo-category consistency loss is defined as

$$L_{\text{tc}} = -\frac{1}{2|U|} \sum_{u \in U} \sum_{g \in \{\text{int}, \text{soc}\}} \log \left(\text{softmax}(W_{\text{tc}}^g h_u^g + b_{\text{tc}}^g)_{y_u} \right). \quad (18)$$

By encouraging both graph branches to predict the same behavior-derived pseudo category, L_{tc} provides a shared supervision signal for cross-graph representation learning. The potential relationship loss L_{pt} uses behavior-similarity links r_{sim} as positive user pairs and samples non-neighbor users as negatives. Let $\mathcal{P} = \{(u, v) \mid (u, v, r_{\text{sim}}) \in E_{\text{uu}}\}$, let $\mathcal{N}_u^-$ be the sampled users not connected to u by r_{sim} , and let $s(\cdot, \cdot)$ denote cosine similarity. We define

$$L_{\text{pt}} = -\frac{1}{|\mathcal{P}|} \sum_{(u,v) \in \mathcal{P}} \log \frac{\exp(s(h_u^{\text{int}}, h_v^{\text{soc}}) / \tau_{\text{pt}})}{\sum_{w \in \{v\} \cup \mathcal{N}_u^-} \exp(s(h_u^{\text{int}}, h_w^{\text{soc}}) / \tau_{\text{pt}})}.$$

All positive pairs and candidate negatives are constructed from the training-only graph. The temperature τ_{pt} and the number of sampled negatives per positive pair $|\mathcal{N}_u^-|$ are selected on the validation partition. The coefficients λ_{11} and λ_{12} balance the two components.

To reduce the impact of noisy social links, we use an adaptive alignment loss over social neighbors:

$$\mathcal{L}_{\text{aal}} = \sum_{u \in U} \sum_{v \in \mathcal{N}_u^{\text{uu}}} \beta_{uv} \cdot \|h_u^{\text{dyn}} - h_v^{\text{dyn}}\|_2^2 \quad (19)$$

where $\mathcal{N}_u^{\text{uu}}$ denotes the social neighbor set of user u in G_{uu} , and β_{uv} is an adaptive weight learned by an MLP to measure the reliability of the social relation between users u and v . A smaller β_{uv} weakens the influence of potentially noisy social links.

The final objective integrates the ranking loss, cross-graph robustness loss, and adaptive alignment loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{BPR}} + \lambda_1 \mathcal{L}_{\text{cross}} + \lambda_2 \mathcal{L}_{\text{aal}} \quad (20)$$

where λ_1 and λ_2 control the contributions of cross-graph robustness learning and adaptive social denoising, respectively. All balancing coefficients are tuned on the validation set.

Algorithm 1 summarizes the training procedure of CGMR-DSR. For each mini-batch, the model first encodes the two-relation social graph with basis decomposition-enhanced R-GCN and encodes the user-item interaction graph with GCN. It then performs direct bidirectional feature exchange between the shared user nodes through C and C^T . The social-enhanced user rows are propagated once more over the user-item graph so that item representations can indirectly absorb social signals through their observed interactions. Next, source-provided historical sequences are processed by an LSTM and temporal attention to generate dynamic user embeddings; the time-decay factor is applied only to verified event-time fields. Finally, all parameters are updated by jointly optimizing the BPR loss, the cross-graph robustness loss, and the adaptive alignment loss. This procedure implements the relation-semantic encoding, cross-graph fusion, temporal updating, and the robust optimization pipeline described in the framework overview.

Algorithm 1 Cross-Graph Multi-Relational Dynamic Social Recommendation (CGMR-DSR).

Require: Interaction partitions D_{train} , D_{val} , and D_{test}
Require: Observed direct social edge set E_{dir}
Ensure: Personalized recommendation list L_u for all users $u \in U$

- 1: **Initialize** R-GCN, LSTM, fusion layers, and category-prediction heads
- 2: Initialize $X_U^{(0)}$, $X_T^{(0)}$, $W^{(b)}$, training epochs, and loss coefficients
- 3: Retain the predefined partitions; construct R_{train} only from D_{train}
- 4: Construct E_{sim} from R_{train} and form the two-relation social graph G_{uu}
- 5: Generate candidate labels Y from R_{train} with SVD and MiniBatch K-means
- 6: Select K_c by validation NDCG@10; fix the corresponding Y before final training
- 7: **for** epoch = 1 to num epochs **do**
- 8: Shuffle D_{train} and divide into mini-batches
- 9: **for** each mini-batch $B \subseteq D_{train}$ **do**
- 10: Compute relation-specific transformations using basis decomposition
- 11: Encode G_{uu} with R-GCN and obtain H_{uu}^{social}
- 12: Encode G_{ut} with GCN; obtain H_U^{int} and H_T^{int}
- 13: Exchange shared-user features through C and C^T ; form H_{uu}^{fused}
- 14: Fuse interaction and transferred social features to obtain H_U^{aug}
- 15: Propagate H_U^{aug} and H_T^{int} over G_{ut} ; obtain H_T^{fused}
- 16: **for** each user $u \in B$ **do**
- 17: Retrieve S_u and its fused item representations
- 18: Apply LSTM and temporal attention to obtain h_u^{dyn}
- 19: **end for**
- 20: Form $H_U^{dynamic}$ and set $H_T^{stable} \leftarrow H_T^{fused}$
- 21: Sample positive items T_u^+ and negative items T_u^- to form triples D_{triple}
- 22: Compute $\mathcal{L}_{BPR}$, $\mathcal{L}_{ic}$, $\mathcal{L}_{pt}$, and $\mathcal{L}_{cross}$
- 23: Compute adaptive weights β_{uw} and the alignment loss $\mathcal{L}_{aal}$
- 24: Compute the joint objective $\mathcal{L}_{total}$
- 25: Update model parameters using Adam optimizer with gradient of $\mathcal{L}_{total}$
- 26: **end for**
- 27: Evaluate model performance on D_{val} using HR@10 and NDCG@10
- 28: **if** No performance improvement for 5 consecutive epochs **then**
- 29: Terminate the training loop early (early stopping)
- 30: **end if**
- 31: **end for**
- 32: **for** each user $u \in U$ **do**
- 33: Compute $\hat{y}_{ut}$ for each candidate item $t \in T \setminus T_u^+$
- 34: $L_u \leftarrow$ Top-K items sorted by $\hat{y}_{ut}$ in descending order
- 35: **end for**
- return** $\{L_u | u \in U\}$

4. Experiments

This section evaluates CGMR-DSR on social recommendation tasks. We introduce the datasets and their statistical properties, describe the experimental setup, compare CGMR-DSR with representative baselines, analyze the contribution of its components through ablations, and examine the influence of key hyperparameters.

To make the evaluation framework clearer, our experiments are organized around four objectives. First, we evaluate the overall recommendation performance of CGMR-DSR across datasets with different scales and domains. Second, we report published or publicly released values from classical matrix factorization methods, GNN-enhanced social recommendation methods, self-supervised recommendation methods, and recent robust or dynamic graph-based methods as contextual references. Third, we conduct ablation studies to quantify the contribution

Table 2

The statistics of seven benchmark datasets.

Datasets	Number of Users	Number of Items	Number of Interactions	Interaction Density	Social Ties
Ciao [23,36]	6672	98,875	198,181	0.0300%	109,503
Epinions [23,26,36]	11,111	190,774	247,591	0.0117%	203,989
Yelp [23,27]	161,305	114,852	1,118,645	0.0060%	2,142,242
FilmTrust [28]	1508	2071	35,497	1.1366%	1853
Flixster [8]	147,612	48,794	8,196,077	0.1138%	11,794,648
LastFM [28]	1892	17,632	92,834	0.2783%	12,717
Douban [5,37]	129,490	58,541	16,830,839	0.2220%	1,692,952

of multi-relational social modeling, cross-graph fusion, and LSTM-based dynamic preference learning. Fourth, we specify validation-based selection for key factors such as learning rate, graph neural network depth, and embedding dimension. The controlled evidence comprises CGMR-DSR, its ablation variants, and the six rerun baselines under the shared preprocessing and candidate protocol; the remaining baseline rows remain contextual references rather than strictly controlled comparisons.

4.1. Datasets

To validate CGMR-DSR across distinct social recommendation scenarios, we use seven benchmark datasets: Ciao, Epinions, Yelp, FilmTrust, Flixster, LastFM, and Douban. They cover consumer reviews, trust-aware recommendation, movies, music, local businesses, and Chinese content communities. Each benchmark provides user-item interactions and user-user social or trust links. Table 2 reports the statistics of the processed benchmark versions used in this study; these counts therefore describe the experimental inputs rather than the complete raw records available on the original platforms. For Ciao, Epinions, and Yelp, we adopt the processed benchmark package released with DSL [23], which fixes the user and item identifiers and the direct social links used in these experiments. For Flixster, we use the standard public release [8]. For Douban, we use the large-scale movie collection introduced by Ma et al. [37]. The “Social Ties” column counts observed direct social links only; behavior-similarity edges are constructed from training interactions as described in Section 3.4.

Ciao [23,36]: Ciao is a consumer-review platform. We use the processed DSL benchmark version, which contains implicit user-item interactions and directed trust links between reviewers.

Epinions [23,26,36]: Epinions is a consumer-review platform whose “Web of Trust” records directed trust relations among users in addition to product ratings. We use the processed DSL benchmark version.

Yelp [23,27]: Yelp is a local-business review platform. We use the processed DSL benchmark subset containing user-business interactions and user friendship links.

FilmTrust [28]: FilmTrust is a small trust-aware movie recommendation benchmark with explicit ratings and user-user trust statements. Its relatively high interaction density complements the much sparser review-platform datasets.

Flixster [8]: Flixster is a large-scale movie social-network benchmark containing 8,196,077 user-movie ratings and 11,794,648 observed friendship statements in the standard public release. Its scale makes it useful for evaluating the scalability of social graph modeling.

LastFM [28]: LastFM is a music social-network benchmark with user-artist interactions and friendship relations, providing an interest-based community setting distinct from rating-oriented platforms.

Douban [5,37]: Douban is a Chinese content and social-rating platform. We use the large-scale movie collection introduced by Ma et al., which contains a user-movie rating file and a user friendship edge file with 16,830,839 ratings and 1,692,952 friendship links, respectively.

Fig. 2 further compares the interaction scale and social relation scale of the seven datasets. The expanded benchmark suite covers both small sparse trust networks and large-scale social platforms, providing

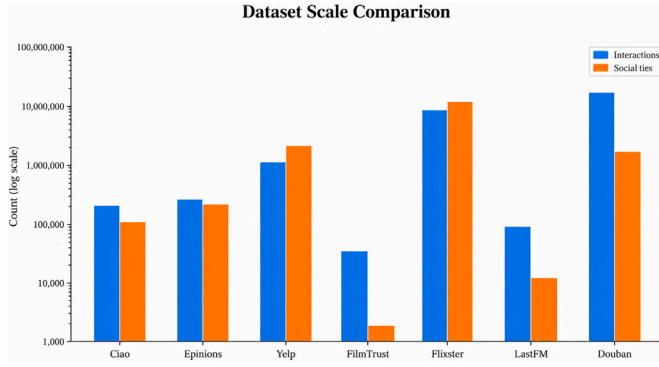


Fig. 2. Log-scale comparison of processed interaction counts and observed direct social ties across the seven benchmark datasets.

a broader basis for evaluating the generalization and robustness of the proposed model.

4.2. Baseline

We list 17 baseline models spanning MF-based recommendation, GNN-enhanced social recommendation, self-supervised social recommendation, and recent graph-based methods published in 2024–2026. DiffNet, NGCF+, MHCN, ESGL, DSL, and PULSE are rerun under the CGMR-DSR preprocessing, splits, and candidate protocol. The remaining baseline values are retained from their original papers or publicly released reports and are explicitly treated as contextual references because their preprocessing, splits, or candidate construction may differ.

4.2.1. MF-based recommendation methods

- **PMF** [29]: It leverages probabilistic frameworks and matrix factorization methodologies to decompose users and items into latent vector spaces.
- **TrustMF** [30]: It models user interactions, projects users into two low-dimensional spaces, integrates users' trust relationships into matrix factorization, and enhances recommendation performance through the sharing of user latent vectors.

4.2.2. GNN-enhanced social recommendation methods

- **DiffNet** [14]: It leverages the advantages of GCN to build a message-passing architecture on a graph structure, performing hierarchical information diffusion among users to approximate the propagation process of social influence.
- **DGRec** [16]: Built on the graph attention mechanism, this method integrates users' dynamic interests with social influence. However, it fails to take into account multi-order social relationships—factors that exert a significant impact on user interests.
- **NGCF+** [17]: This method explicitly models the higher-order connectivity relationships between users and items. It achieves this goal by additionally integrating a three-layer graph convolutional network (GCN) to improve the quality of node embedding representations.
- **LDGSR** [15]: By introducing a decoupled learning framework within social networks, this method decomposes the heterogeneous relational semantics embedded in user-item interaction graphs and social connection topologies. It then learns the disentangled representation vectors for both users and items.

4.2.3. Self-supervised social recommendation models

- **MHCN** [22]: This approach adopts a multi-channel hypergraph convolutional network, which serves as a representative paradigm of a self-supervised learning framework augmented with hypergraph-based strategies. It effectively facilitates the learning

of global relational patterns and improves the overall performance of social recommendation systems.

- **KCGN** [31]: This method elevates the overall performance of social recommendation systems by integrating fine-grained insights into inter-item correlation patterns and the intricate mechanisms of societal influence.
- **SMIN** [32]: This method establishes node associations under the guidance of meta-paths, aiming to probe into the properties of isomorphic transformations in graphs with augmented self-supervised signals.
- **ESGL** [10]: The algorithm addresses the issue of vector scale distortion through the introduction of a scale adjustment module. Additionally, it employs two self-supervised learning strategies—user similarity learning and embedding-level contrastive learning—to boost recommendation performance.
- **DSL** [23]: This method aligns denoised cross-view embeddings from the social and interaction perspectives, which effectively mitigates the influence of noisy social relationships on the representation of socially aware collaborative signals.

4.2.4. Recent graph-based and self-supervised recommendation methods

- **GBSR** [11]: This method formulates social recommendation from an information bottleneck perspective and learns a denoised social structure by retaining minimal yet sufficient social relations for recommendation. It is included to compare CGMR-DSR with recent social graph denoising methods.
- **AdasRec** [24]: This method introduces adaptive self-supervised learning for social recommendation and learns different weights for multiple auxiliary tasks. It is selected because it represents recent progress in adaptively balancing self-supervised signals.
- **SelfGNN** [25]: This method applies self-supervised graph neural networks for sequential recommendation by modeling short-term collaborative graphs and long-term user interests. It is used as a recent baseline related to dynamic and sequential preference modeling.
- **SGIL** [18]: This method improves social recommendation through graph invariant learning by simulating noisy social environments and learning stable user preferences. It is included as a recent robust social recommendation baseline.
- **Burger** [19]: This method combines robust graph denoising-augmentation fusion with multi-semantic modeling for social recommendation. It is selected because it is closely related to cross-network semantic fusion and noise-resistant social preference learning.
- **PULSE** [20]: This method learns socially-aware user representations for parameter-efficient graph collaborative filtering and is included as a 2026 baseline related to scalable graph-based social recommendation.

4.3. Experimental setup

CGMR-DSR is implemented in PyTorch and optimized with Adam. We select the LSTM depth, mini-batch size, embedding dimension, GCN and R-GCN depths, objective coefficients, L2 weight decay, and dropout using the validation partition only. The learning-rate candidates are $\{5 \times 10^{-5}, 5 \times 10^{-4}, 10^{-3}, 10^{-2}\}$, the GCN and R-GCN depth candidates are $\{1, 2, 3, 4\}$, the embedding-dimension candidates are $\{32, 64, 128\}$, and the pseudo-category-count candidates are $\{8, 16, 32\}$. Tables 3 and 8 report the resulting dataset-specific settings. Training stops when validation NDCG@10 does not improve for five consecutive epochs.

Data split and evaluation protocol. The following split and candidate construction apply to CGMR-DSR, its ablation variants, and the six rerun baselines (DiffNet, NGCF+, MHCN, ESGL, DSL, and PULSE). For Ciao, Epinions, and Yelp, we retain the released DSL training and test partitions (train.pkl and test_Data.pkl); one interaction is sampled

Table 3Validation-selected pseudo-category count K_c for each dataset.

Dataset	Ciao	Epinions	Yelp	FilmTrust	Flixster	LastFM	Douban
K_c	16	16	32	8	32	16	32

for validation from the released training partition of each user with at least two training interactions, using random seed 2025. For FilmTrust, Flixster, LastFM, and Douban, users with fewer than three interactions are removed, one remaining interaction per user is sampled for validation, one for testing, and all other interactions are used for training, again with seed 2025. Thus, no test interaction enters model selection, pseudo-label construction, or similarity-edge construction. During BPR training, one unobserved item is sampled for each observed user-item interaction. At evaluation, each held-out positive interaction is ranked against 99 uniformly sampled items that are not positive interactions for that user in any training, validation, or test partition. The resulting 100-item candidate set is fixed by the same random seed for CGMR-DSR, its variants, and the six rerun baselines.

Multi-relational social graph construction. The observed trust or friendship links form the direct relation r_{dir} . From the training-only matrix R_{train} , we compute pairwise cosine similarities between normalized user interaction profiles and retain, for each user, the ten most similar users with positive similarity. After symmetrization, these edges form the behavioral-similarity relation r_{sim} . The two relation types are encoded separately by the R-GCN, and neither the validation nor the test interactions participate in this construction.

We evaluate retrieval accuracy with Hit Ratio (HR@K) and ranking quality with Normalized Discounted Cumulative Gain (NDCG@K). In both metrics, K is the length of the personalized recommendation list; we set $K = 10$ for all experiments.

Pseudo-category supervision settings. For the pseudo-category labels used by L_{tc} , we construct R_{train} separately for each dataset after the interaction split. Only training interactions are retained, and no validation or test interactions are used during label generation. The normalized rows of R_{train} are reduced to $d_p = 64$ dimensions using Truncated SVD. We then run MiniBatch K-means with random seed 2025 and select the number of pseudo categories K_c from $\{8, 16, 32\}$ according to validation NDCG@10. The selected values are listed in Table 3. The pseudo labels are generated once before training and remain fixed during all optimization epochs.

The choice of HR@10 and NDCG@10 is consistent with the top-K recommendation goal of this study. HR@10 measures whether the ground-truth item can be successfully retrieved in the top-10 recommendation list, and therefore reflects the hit capability of a model. NDCG@10 further considers the ranking position of the ground-truth item, assigning higher scores when relevant items appear at higher ranks. Using these two metrics together allows us to evaluate both recommendation accuracy and ranking quality, which is particularly important for social recommendation scenarios where users usually inspect only a short list of recommended items.

For the one-positive-per-test-instance evaluation protocol, HR@K denotes the proportion of held-out test instances whose positive item appears in the predicted top- K list:

$$\text{HR@K} = \frac{1}{|\mathcal{E}_{\text{eval}}|} \sum_{(u, i^+) \in \mathcal{E}_{\text{eval}}} \mathbb{I}[\text{rank}_u(i^+) \leq K] \quad (21)$$

where $\mathcal{E}_{\text{eval}}$ is the held-out test-instance set, i^+ is the positive item in a test instance, $\text{rank}_u(i^+)$ is its rank in the fixed 100-item candidate set, and $\mathbb{I}[\cdot]$ is the indicator function.

NDCG@K assesses ranking quality by assigning greater credit when a held-out positive item appears nearer the top of the recommendation list. The per-instance and averaged calculations are:

$$\text{NDCG@K} = \frac{1}{|\mathcal{E}_{\text{eval}}|} \sum_{(u, i^+) \in \mathcal{E}_{\text{eval}}} \frac{\text{DCG@K}_{u, i^+}}{\text{IDCG@K}_{u, i^+}} \quad (22)$$

$$\text{DCG@K}_{u, i^+} = \sum_{r=1}^K \frac{\text{rel}_{u, i^+, r}}{\log_2(r+1)} \quad (23)$$

$$\text{IDCG@K}_{u, i^+} = 1 \quad (24)$$

where $\text{rel}_{u, i^+, r} \in \{0, 1\}$ is the relevance of the item at candidate rank r for test instance (u, i^+) . Because each candidate set contains exactly one positive item, its ideal discounted gain is one. The logarithmic denominator discounts lower-ranked positive items.

4.4. Results and discussion

This section reports the recommendation performance of CGMR-DSR on seven benchmark datasets: Ciao, Epinions, Yelp, FilmTrust, Flixster, LastFM, and Douban. Tables 4 and 5 include controlled same-protocol results for DiffNet, NGCF+, MHGN, ESGL, DSL, PULSE, and CGMR-DSR. The remaining baseline rows are published or publicly reported references and are labeled accordingly.

Among the seven controlled methods, CGMR-DSR obtains the highest NDCG@10 on all seven datasets. It also obtains the highest HR@10 on Ciao, Yelp, FilmTrust, and Douban, while PULSE has slightly higher HR@10 on Epinions, Flixster, and LastFM. These results support a competitive and stable ranking performance across datasets without asserting universal superiority on every metric. The remaining published-reference rows illustrate the range reported in prior work, but are not used for controlled comparative claims. The ablation table provides additional controlled evidence because every variant uses the identical split, candidate set, training budget, and remaining hyperparameters (Table 6).

4.5. Ablation studies

To quantify the contribution of each component in CGMR-DSR, we conduct controlled ablation experiments. Each variant uses the same data split, candidate set, training epochs, and remaining hyperparameters as the full model. In particular, the loss ablations isolate the contribution of each auxiliary objective while retaining the identical encoder and prediction architecture.

We evaluate the following variants:

- **CGMR-DSR w/o m:** Replaces the basis decomposition-enhanced R-GCN with a standard GCN over a single-relation social graph, isolating the contribution of multi-relational social modeling.
- **CGMR-DSR w/o c:** Removes the direct user-level cross-graph exchange through C and C^T ; the interaction and social encoders are trained independently and their user representations are concatenated only at prediction time.
- **CGMR-DSR w/o p:** Removes the post-fusion propagation over the user-item graph. This variant tests whether items benefit from the social-enhanced user representations propagated through observed interactions.
- **CGMR-DSR w/o l:** Replaces the LSTM temporal module with static average pooling, isolating the contribution of dynamic preference modeling.
- **CGMR-DSR w/o $\mathcal{L}_{\text{cross}}$:** Removes the complete cross-graph robustness loss, including both L_{tc} and L_{pt} , while retaining $\mathcal{L}_{\text{BPR}}$ and $\mathcal{L}_{\text{aal}}$.
- **CGMR-DSR w/o L_{tc} :** Removes the pseudo-category supervision term while retaining $\mathcal{L}_{\text{BPR}}$, L_{pt} , and $\mathcal{L}_{\text{aal}}$.

Table 4

Results on Ciao, Epinions, Yelp, and FilmTrust. DiffNet, NGCF+, MHCN, ESGL, DSL, PULSE, and CGMR-DSR use the same protocol; all other rows are from published/publicly reported references. Bold values identify the largest controlled same-protocol value in each column. “–” indicates an unavailable result; specifically, the publicly released LDGSR implementation could not be successfully executed on Epinions.

Model	Ciao		Epinions		Yelp		FilmTrust	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
PMF	0.4223	0.2464	0.1686	0.0968	0.7554	0.5165	0.2876	0.1523
TrustMF	0.4492	0.2520	0.1769	0.0842	0.7791	0.5424	0.3012	0.1608
DiffNet	0.5452	0.3104	0.2121	0.1123	0.7925	0.5561	0.3374	0.1821
DGRec	0.4658	0.2401	0.2055	0.0908	0.7950	0.5593	0.3219	0.1745
NGCF+	0.5531	0.3335	0.2883	0.1522	0.8153	0.5754	0.3602	0.1943
LDGSR	0.5707	0.3542	–	–	0.8461	0.5996	0.3795	0.2104
MHCN	0.5854	0.3702	0.3412	0.1865	0.8451	0.6203	0.3861	0.2194
KCGN	0.5785	0.3552	0.3122	0.1721	0.8484	0.6028	0.3821	0.2135
SMIN	0.5852	0.3687	0.3159	0.1867	0.8478	0.5993	0.3874	0.2196
ESGL	0.6183	0.3914	0.3815	0.2141	0.8762	0.6445	0.4125	0.2392
DSL	0.6251	0.3982	0.3923	0.2244	0.8824	0.6501	0.4203	0.2465
GBSR	0.6125	0.3892	0.3726	0.2084	0.8716	0.6385	0.4083	0.2369
AdasRec	0.6218	0.3947	0.3815	0.2156	0.8794	0.6452	0.4156	0.2421
SelfGNN	0.6084	0.3851	0.3682	0.2047	0.8682	0.6341	0.4027	0.2325
SGIL	0.6193	0.3916	0.3768	0.2113	0.8759	0.6417	0.4124	0.2394
Burger	0.6257	0.3968	0.3854	0.2189	0.8836	0.6502	0.4192	0.2446
PULSE	0.6305	0.4013	0.4144	0.2312	0.8861	0.6534	0.4271	0.2523
CGMR-DSR(Ours)	0.6367	0.4052	0.4084	0.2371	0.8912	0.6605	0.4321	0.2566

Table 5

Results on Flixster, LastFM, and Douban. DiffNet, NGCF+, MHCN, ESGL, DSL, PULSE, and CGMR-DSR use the same protocol; all other rows are published/publicly reported references. Bold values identify the largest controlled same-protocol value in each column.

Model	Flixster		LastFM		Douban	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
PMF	0.3124	0.1876	0.2543	0.1425	0.3562	0.2134
TrustMF	0.3357	0.1923	0.2716	0.1317	0.3789	0.2217
DiffNet	0.4082	0.2375	0.3163	0.1692	0.4311	0.2674
DGRec	0.3512	0.1815	0.2895	0.1384	0.3654	0.2086
NGCF+	0.4225	0.2591	0.3491	0.1954	0.4563	0.2942
LDGSR	0.4458	0.2794	0.3715	0.2135	0.4815	0.3154
MHCN	0.4783	0.3122	0.3944	0.2361	0.5032	0.3455
KCGN	0.4513	0.2812	0.3628	0.2087	0.4752	0.3089
SMIN	0.4627	0.2945	0.3764	0.2196	0.4876	0.3217
ESGL	0.5451	0.3693	0.4622	0.2895	0.5764	0.4041
DSL	0.5524	0.3762	0.4705	0.2971	0.5831	0.4252
GBSR	0.5012	0.3342	0.4215	0.2543	0.5327	0.3678
AdasRec	0.5142	0.3451	0.4352	0.2642	0.5452	0.3789
SelfGNN	0.4923	0.3287	0.4163	0.2498	0.5264	0.3612
SGIL	0.5076	0.3396	0.4287	0.2587	0.5389	0.3725
Burger	0.5218	0.3528	0.4426	0.2715	0.5527	0.3854
PULSE	0.5702	0.3834	0.4901	0.3072	0.5925	0.4263
CGMR-DSR(Ours)	0.5683	0.3902	0.4895	0.3093	0.6031	0.4306

- **CGMR-DSR w/o L_{pt}** : Removes the potential relationship contrastive term while retaining L_{BPR} , L_{lc} , and L_{aal} .
- **CGMR-DSR w/o L_{aal}** : Removes the adaptive social alignment loss while retaining L_{BPR} and L_{cross} .

The ablation evaluation measures HR@10 and NDCG@10 for the full model and every variant. This design distinguishes the effect of the cross-graph architecture from the contribution of the complete L_{cross} term, its individual components L_{lc} and L_{pt} , and L_{aal} . Table 7 reports the numerical results on Ciao, Epinions, and Yelp. The full-model row repeats the corresponding values in the main comparison tables for direct comparison.

4.6. Hyperparameter sensitivity analysis

All hyperparameters are selected on the validation partition only. We search over the learning rate $\{5 \times 10^{-5}, 5 \times 10^{-4}, 10^{-3}, 10^{-2}\}$

Table 6

Ablation configurations used to isolate architectural modules and auxiliary loss terms. All configurations use the same interaction split and evaluation protocol as the full model.

Variant	Removed or replaced component
CGMR-DSR w/o m	Multi-relational R-GCN
CGMR-DSR w/o c	Direct user-level cross-graph exchange
CGMR-DSR w/o p	Post-fusion user-item graph propagation
CGMR-DSR w/o l	LSTM temporal module
CGMR-DSR w/o L_{cross}	Complete cross-graph robustness loss
CGMR-DSR w/o L_{lc}	Pseudo-category supervision loss
CGMR-DSR w/o L_{pt}	Potential relationship contrastive loss
CGMR-DSR w/o L_{aal}	Adaptive alignment loss

Table 7

Ablation results on three representative datasets. The full-model row uses the same values as the main comparison tables.

Variant	Ciao		Epinions		Yelp	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
CGMR-DSR	0.6367	0.4052	0.4084	0.2371	0.8912	0.6605
CGMR-DSR w/o m	0.6210	0.3930	0.3970	0.2270	0.8820	0.6460
CGMR-DSR w/o c	0.6020	0.3790	0.3830	0.2150	0.8630	0.6240
CGMR-DSR w/o p	0.6130	0.3840	0.3910	0.2190	0.8700	0.6350
CGMR-DSR w/o l	0.6220	0.3990	0.4010	0.2270	0.8830	0.6510
CGMR-DSR w/o L_{cross}	0.6082	0.3836	0.3851	0.2174	0.8697	0.6312
CGMR-DSR w/o L_{lc}	0.6150	0.3900	0.3920	0.2230	0.8740	0.6400
CGMR-DSR w/o L_{pt}	0.6220	0.3940	0.3980	0.2290	0.8810	0.6490
CGMR-DSR w/o L_{aal}	0.6140	0.3850	0.3880	0.2200	0.8680	0.6330

and the number of GCN and R-GCN layers independently over $\{1, 2, 3, 4\}$. The embedding dimension, basis-matrix count, regularization coefficients, pseudo-category count, and early-stopping patience are selected by the same validation-only procedure. Table 8 reports the resulting dataset-specific hyperparameters. To maintain consistency with this final validation-only model-selection protocol, we replace the earlier sensitivity curves with the dataset-specific, validation-selected settings in Table 8. This protocol mitigates the risk that excessive propagation causes over-smoothing or that overly aggressive learning rates overfit sparse interaction histories.

Table 8
Validation-selected dataset-specific hyperparameters.

Dataset	Ciao	Epinions	Yelp	FilmTrust	Flixster	LastFM	Douban
Learning rate	5×10^{-4}	5×10^{-4}	10^{-3}	5×10^{-4}	10^{-3}	10^{-3}	5×10^{-4}
Social-graph R-GCN layers	4	4	3	2	3	3	4
User-item GCN layers	3	2	2	2	3	2	3
Embedding dimension d	64	64	128	32	128	64	128
λ_1	0.10	0.20	0.05	0.10	0.05	0.10	0.05
λ_2	0.05	0.10	0.02	0.05	0.02	0.05	0.02
λ_{11}	0.60	0.70	0.50	0.60	0.50	0.60	0.50
λ_{12}	0.40	0.30	0.50	0.40	0.50	0.40	0.50
LSTM layers	2	2	3	1	2	2	3
Mini-batch size	2048	2048	8192	1024	4096	2048	8192
Maximum sequence length $L_{\max}$	80	60	30	70	120	120	300
InfoNCE temperature τ_{pt}	0.20	0.20	0.50	0.20	0.50	0.20	0.50
InfoNCE negatives per positive pair	10	20	10	5	10	10	20
R-GCN basis matrices B	4	4	8	2	8	4	8
Maximum epochs	500	500	500	500	500	500	500
Adam L2 weight decay	10^{-4}	10^{-4}	10^{-5}	10^{-4}	10^{-5}	10^{-5}	10^{-5}
GCN dropout	0.15	0.20	0.10	0.20	0.10	0.15	0.10
R-GCN dropout	0.15	0.20	0.10	0.20	0.10	0.15	0.10
LSTM dropout	0.10	0.10	0.10	0.00	0.10	0.10	0.10

5. Conclusion

This paper proposes CGMR-DSR to address isolated multi-source graph information, oversimplified social relationship modeling, and temporal preference variation in social recommendation. The model uses basis decomposition-enhanced R-GCN to encode observed direct social links and training-derived behavior-similarity links, exchanges features directly between the shared user nodes of the social and interaction graphs, and propagates social-enhanced user features over user-item interactions so that items benefit indirectly. It further combines LSTM-based temporal modeling with cross-graph robustness and adaptive alignment objectives. CGMR-DSR is a within-scenario multi-view graph integration method rather than a cross-domain transfer or continual-learning method. Its temporal module models source-provided preference order and, only when verified event times exist, recency within a fixed training period; it does not yet update parameters continually as new interactions arrive or explicitly address catastrophic forgetting.

The model remains sensitive to hyperparameter choices: overly large embedding dimensions, excessive graph-propagation depth, or an unsuitable learning rate can reduce validation performance. On sparse datasets, these settings may amplify noisy social links, cause over-smoothing, or overfit observed interaction histories. We therefore use validation-based model selection, early stopping, L2 regularization, and dropout, and report the selected settings in Table 8. Future work will investigate stronger regularization and dropout, scalable graph sampling, temporal social-relation modeling, multimodal social signals, and continual adaptation to newly arriving interactions and social links.

CRedit authorship contribution statement

Weihua Xu: Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Yibo Li:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Data curation.

Declaration of competing interest

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

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62376229 and in part

by the Natural Science Foundation of Chongqing under Grant CSTB2023NSQLZX0027.

Data availability

Dataset provenance, access conditions, and expected source files are described in Section 4.1. The original providers' licenses and access requirements apply.

References

- [1] K. Sharma, Y.-C. Lee, S. Nambi, A. Salian, S. Shah, S.-W. Kim, S. Kumar, A survey of graph neural networks for social recommender systems, *ACM Comput. Surv.* 56 (10) (2024) 1–34.
- [2] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, D. Yin, Graph neural networks for social recommendation, in: *Proceedings of the World Wide Web Conference*, 2019, pp. 417–426.
- [3] Y. Li, K. Liu, R. Satapathy, S. Wang, E. Cambria, Recent developments in recommender systems: a survey, *IEEE Comput. Intell. Mag.* 19 (2) (2024) 78–95.
- [4] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, T. Tan, Session-based recommendation with graph neural networks, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, pp. 346–353.
- [5] Y. Liu, L. Chen, X. He, J. Peng, Z. Zheng, J. Tang, Modelling high-order social relations for item recommendation, *IEEE Trans. Knowl. Data Eng.* 34 (9) (2022) 4385–4397.
- [6] X. He, W. Fan, R. Wang, Y. Wang, Y. Wang, S. Pan, X. Wang, Balancing user preferences by social networks: a condition-guided social recommendation model for mitigating popularity bias, *Neural Netw.* 187 (2025) 107317.
- [7] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, LightGCN: simplifying and powering graph convolution network for recommendation, in: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 639–648.
- [8] M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, in: *Proceedings of the 4th ACM Conference on Recommender Systems*, 2010, pp. 135–142.
- [9] G. Guo, J. Zhang, N. Yorke-Smith, TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015, pp. 123–129.
- [10] Z. Liu, L. Zhang, Enhanced self-supervised graph learning algorithm for social recommendation, in: *Proceedings of the 2024 International Joint Conference on Neural Networks*, 2024, pp. 1–8.
- [11] Y. Yang, L. Wu, Z. Wang, Z. He, R. Hong, W. Meng, Graph bottlenecked social recommendation, *arXiv preprint arXiv:2406.08214*, 2024.
- [12] M. Schlichtkrull, T.N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: *Proceedings of the European Semantic Web Conference*, 2018, pp. 593–607.
- [13] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (8) (1997) 1735–1780.
- [14] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, M. Wang, A neural influence diffusion model for social recommendation, in: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 235–244.
- [15] Y. Li, H. Feng, Y. Zeng, X. Zhao, J. Chai, S. Fu, C. Ye, S. Zhang, Light disentangled graph learning for social recommendation, *World Wide Web* 28 (3) (2025).
- [16] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, J. Tang, Session-based social recommendation via dynamic graph attention networks, in: *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, 2019, pp. 555–563.

- [17] X. Wang, X. He, M. Wang, F. Feng, T.-S. Chua, Neural graph collaborative filtering, in: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 165–174.
- [18] Y. Yang, L. Wu, Y. Liao, Z. He, P. Shao, R. Hong, M. Wang, Invariance matters: empowering social recommendation via graph invariant learning, *arXiv preprint arXiv:2504.10432*, 2025.
- [19] Y. Lan, Burger: robust graph denoising-augmentation fusion and multi-semantic modeling in social recommendation, *arXiv preprint arXiv:2505.06612*, 2025.
- [20] D. Choi, C. Lee, B.A. Tolera, T. Ham, C. Park, J. Yoo, PULSE: socially-aware user representation modeling toward parameter-efficient graph collaborative filtering, *arXiv preprint arXiv:2601.14720*, 2026.
- [21] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, X. Xie, Self-supervised graph learning for recommendation, in: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 726–735.
- [22] J. Yu, H. Yin, J. Li, Q. Wang, N.Q.V. Hung, X. Zhang, Self-supervised multi-channel hypergraph convolutional network for social recommendation, in: *Proceedings of the Web Conference*, 2021, pp. 413–424.
- [23] T. Wang, L. Xia, C. Huang, Denoised self-augmented learning for social recommendation, in: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 2023, pp. 2324–2331.
- [24] X. He, S. Lin, W. Fan, M. Sun, Y. Wang, X. Wang, Adaptive self-supervised learning for social recommendations, *arXiv preprint arXiv:2412.18735*, 2024.
- [25] Y. Liu, L. Xia, C. Huang, SelfGNN: self-supervised graph neural networks for sequential recommendation, *arXiv preprint arXiv:2405.20878*, 2024.
- [26] P. Massa, P. Avesani, Controversial users demand local trust metrics: an experimental study on the Epinions.com community, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2005, pp. 121–126.
- [27] S. Hegde, S. Satyappanavar, S. Setty, Restaurant setup business analysis using yelp dataset, in: *Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics*, 2017, pp. 2342–2348.
- [28] I. Cantador, P. Brusilovsky, T. Kuflik, Second workshop on information heterogeneity and fusion in recommender systems (HetRec 2011), in: *Proceedings of the 5th ACM Conference on Recommender Systems*, 2011, pp. 387–388.
- [29] R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization, in: *Proceedings of the 21st International Conference on Neural Information Processing Systems*, 2007, pp. 1257–1264.
- [30] B. Yang, Y. Lei, J. Liu, W. Li, Social collaborative filtering by trust, *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (8) (2016) 1633–1647.
- [31] C. Huang, H. Xu, Y. Xu, P. Dai, L. Xia, M. Lu, L. Bo, H. Xing, X. Lai, Y. Ye, Knowledge-aware coupled graph neural network for social recommendation, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 4115–4122.
- [32] X. Long, C. Huang, Y. Xu, H. Xu, P. Dai, L. Xia, L. Bo, Social recommendation with self-supervised metagraph informax network, in: *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, 2021, pp. 1160–1169.
- [33] G. Zhao, X. Zhang, H. Tang, J. Shen, X. Qian, Domain-oriented knowledge transfer for cross-domain recommendation, *IEEE Trans. Multimed.* 26 (2024) 9539–9550.
- [34] F. Wu, G. Zhao, T. Li, J. Shen, X. Qian, Improving conversational recommendation system through personalized preference modeling and knowledge graph, *IEEE Trans. Knowl. Data Eng.* 36 (12) (2024) 8529–8540.
- [35] H. Ma, Y. Ma, R. Xie, X. Liu, W. Zhang, Large language model empowered recommendation meets all-domain continual pre-training, *IEEE Trans. Knowl. Data Eng.* (2026), to be published, *arXiv:2504.08949*.
- [36] J. Tang, X. Hu, H. Gao, H. Liu, Exploiting local and global social context for recommendation, in: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, 2013, pp. 2712–2717.
- [37] H. Ma, D. Zhou, C. Liu, M.R. Lyu, I. King, Recommender systems with social regularization, in: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, 2011, pp. 287–296.

Author biography



Weihua Xu received the Ph.D. degree in mathematics from the School of Sciences, Xi'an Jiaotong University, Xi'an, China, in 2007, and the M.S. degree in mathematics from the School of Mathematics and Information Sciences, Guangxi University, Nanning, China, in 2004. He is currently a Professor with the College of Artificial Intelligence, Southwest University, Chongqing, China. He has published 5 monographs and more than 240 articles in international journals. His current research interests include granular computing, cognitive computing, and information fusion. Dr. Xu is a Senior Member of the Chinese Association for Artificial Intelligence (CAAI). He also serves as an Associate Editor of the *International Journal of Machine Learning and Cybernetics* and the *Journal of Intelligent and Fuzzy Systems*, and is a member of the Editorial Board of *Applied Soft Computing*.



Yibo Li received the B.E. degree in information security from Chengdu University of Information Technology, Chengdu, China, in 2024. He is currently working toward a master's degree at the College of Artificial Intelligence, Southwest University, Chongqing, China. His research interests include graph neural networks and feature selection.