Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

RIBGCL: Reliability-Guided Item-Bias-Aware Graph Contrastive Learning for Social Recommendation

Yinqi Zhang
Shanghai DianJi University
Peilin Yang
peilin_yang@sdju.edu.cn
Shanghai DianJi University
Xiaogang Wang
Shanghai DianJi University
Ruixu Wu
Shanghai DianJi University

Research Article

Keywords: social recommendation, graph contrastive learning, social confidence gating, representation refinement, popularity bias

Posted Date: November 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11487791/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

RIBGCL: Reliability-Guided Item-Bias-Aware Graph Contrastive Learning for Social Recommendation

Yinqi Zhang[1], Peilin Yang[1*], Xiaogang Wang[1], Ruixu Wu[1]

[1*]School of Artificial Intelligence and Data Science, Shanghai DianJi University, Shanghai, China.

*Corresponding author(s). E-mail(s): peilin_yang@sdju.edu.cn;

Abstract

Noisy social ties and popularity-skewed interactions can weaken graph-based social recommendation by transferring irrelevant preferences and overemphasizing frequently observed items. We propose RIBGCL (Reliability-Guided Item-Bias-Aware Graph Contrastive Learning), a framework that coordinates confidence-weighted social propagation with a bounded item-representation correction. First, an interest-consistency gate estimates the relative compatibility of socially connected users and reweights their messages while retaining residual social connectivity. Second, an item-neighborhood branch derives a normalized correction direction and applies similarity-controlled, weak residual refinement to item embeddings. These two components target complementary sources of representation contamination and are trained jointly with social and interaction objectives and augmentation-free contrastive learning. Analytically, the gate preserves a positive lower bound on observed social-edge weights, while the unit-normalized item correction bounds the perturbation of each predicted score. On Epinions, Ciao, and FilmTrust, RIBGCL improves the reported Top-10 ranking metrics over the evaluated baselines; component ablations and social-edge filtering comparisons support the usefulness of both modules. Because the item correction is derived from observational interactions, it is treated as counterfactual-inspired rather than as an identified causal effect.

Keywords: social recommendation, graph contrastive learning, social confidence gating, representation refinement, popularity bias

1 Introduction

Recommender systems play a significant role in filtering content that aligns with user interests from massive information, and are widely applied in scenarios such as e-commerce, short-video platforms, and online social media (Huang et al. 2025a; Zhang et al. 2025c). Traditional collaborative filtering methods primarily model user preferences based on historical interaction behaviors, generating personalized recommendations by mining the associations between users and items (Rendle et al. 2009; Hu et al. 2008). However, in real-world applications, user interaction data are typically very sparse, and some new or low-activity users lack sufficient historical behavior records, making it difficult for models to accurately characterize their interest preferences (Rodríguez-Baena et al. 2025). To alleviate data sparsity and cold-start problems, social recommendation further incorporates social relationships among users, leveraging friendship, trust, or follow connections to supplement user behavioral information and thereby improve recommendation performance (Chen et al. 2019a; Ma et al. 2011).

In recent years, as graph neural networks (GNNs) have been increasingly applied to recommendation tasks, researchers have begun to model user–item interaction relationships and user–user social relationships as graph structures, capturing high-order collaborative signals and social influence through graph propagation mechanisms (Wang et al. 2019; Anand and Maurya 2025; Gao et al. 2023). Compared with early matrix factorization-based social recommendation methods, GNNs are capable of propagating information across multi-hop neighborhoods, making fuller use of the complex associations between users and items (He et al. 2020; Wang et al. 2019; Pires et al. 2026). More recent graph contrastive recommendation models use adaptive propagation and augmentation-free representation alignment to reduce the dependence on random structural perturbations, while keeping multi-hop collaborative information accessible (Wu et al. 2026).

Despite the progress made by existing methods in graph propagation and contrastive learning, two noteworthy issues remain in real-world social recommendation scenarios. First, social relations are not always reliable. The existence of a social connection between two users does not necessarily imply that they share consistent interest preferences; some weakly correlated, accidentally formed, or interest-divergent social edges may introduce noisy information during message propagation, thereby interfering with target user representation learning (Hu et al. 2025; Zhang et al. 2025a; Li and Yang 2025). Existing social recommendation methods typically propagate directly on the original social graph and lack adaptive assessment of social edge reliability, making it difficult to distinguish between beneficial and noisy social neighbors (Hu et al. 2025; Li and Yang 2025; Yang et al. 2025a). Taking trust relationships on e-commerce platforms as an example, follow or trust connections between users are not necessarily driven entirely by similarity in consumption interests. Some users may establish connections due to offline acquaintances, a single transaction review, temporary interactions, or social courtesy, yet their long-term consumption preferences may differ substantially. As shown in Fig. 1, user A consistently browses and purchases digital accessories and gaming equipment, while user B, whom A trusts, primarily focuses on maternal and infant products and household goods. If the model

indiscriminately aggregates B's behavioral information during social graph propagation, maternal or household preferences may be erroneously propagated into A's user representation, causing the model's characterization of A's interests to deviate. Such phenomena indicate that the existence of a social edge can only signify that some form of social relationship exists between users, and cannot be directly equated with interest similarity; therefore, adaptive assessment of social edge reliability is necessary.

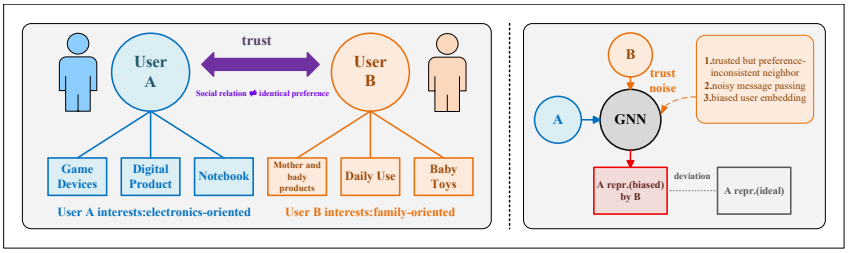


Fig. 1 Social relation noise causes preference shift in an e-commerce trust graph.

Second, item bias is pervasive in user–item interaction data. Popular items receive more exposure opportunities and are more likely to be interacted with by a large number of users; during training, models may over-learn these high-frequency item signals, causing item representations to incorporate non-personalized factors such as popularity (Chen et al. 2021b; Yang et al. 2023; Li et al. 2026b). This bias weakens the model's ability to characterize users' genuine interests, and in sparse data scenarios, models are particularly prone to relying on popular items to form recommendation results. As shown in Fig. 2, using movie recommendation as an example, popular movies typically enjoy higher exposure and more historical ratings, making them more likely to generate high-frequency interaction signals in training data. If a model directly learns item representations based on interaction frequency, it may mistake "being watched by a large number of users" for "being suitable for most users," thereby tending to repeatedly recommend the same batch of popular movies to users with diverse interests. For users who prefer niche documentaries or independent films, such popularity signals obscure their genuine interests and render recommendation results lacking in personalization. Therefore, merely improving graph propagation capability is insufficient to guarantee the reliability of recommendation representations; it is also necessary to further suppress the influence of item-side bias on user preference modeling.

To address these coupled reliability problems, RIBGCL uses two complementary representation controls. For social propagation, an interest-consistency gate transforms user similarity into residual edge weights, allowing the model to attenuate less compatible neighbors without discarding all social connectivity. For item modeling, a neighborhood-derived auxiliary representation provides a direction for bounded residual refinement, avoiding unrestricted subtraction from the primary collaborative embedding. The two controls are incorporated into the same graph contrastive learning objective, permitting interest-domain and social-domain signals to interact during training. Our contribution lies in integrating two explicitly controlled operations at

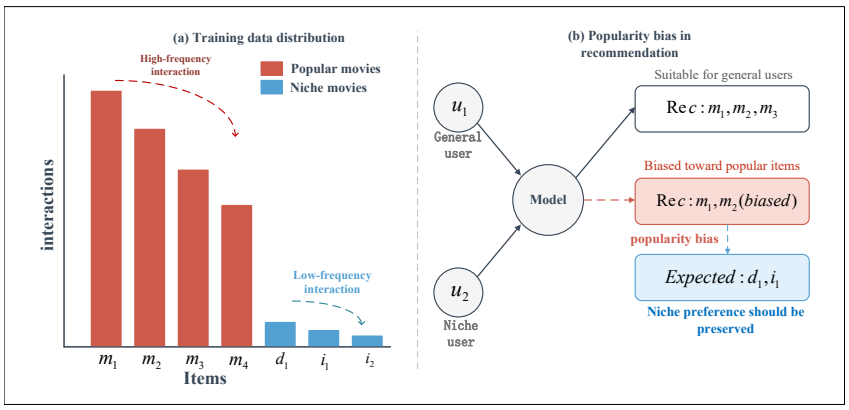


Fig. 2 Popularity bias causes niche users to receive non-personalized recommendations.

the social-edge and item-embedding levels. Unlike an unrestricted item transformation, the refinement admits a direct perturbation bound; unlike hard filtering, the gate leaves every observed social edge with positive residual weight. These are mathematical properties of the stated formulation, not guarantees of uniformly better recommendation accuracy. The item-side direction is an observational proxy rather than a formally identified causal effect.

The main contributions of this paper are summarized as follows:

- We introduce an interest-consistency-aware social propagation mechanism that assigns bounded, residual confidence weights to observed social edges. This offers a soft alternative to hard edge filtering and provides a targeted way to regulate uncertain neighbor messages within a graph contrastive recommender.
- We develop a neighborhood-guided, bias-aware item refinement mechanism. Its learned normalized direction and similarity-dependent correction coefficient bound the change to each item embedding, providing a conservative way to reduce potentially non-personalized components without removing the primary collaborative signal.
- We integrate these mechanisms into a unified social recommendation framework and evaluate their complementary effects on three benchmark datasets. Reported improvements, component ablations, and comparisons with hard social-edge filtering characterize the benefits and remaining limitations of the approach.

2 Related Work

2.1 General Recommendation

The core of general recommendation lies in inferring users' preferences for items from their historical behaviors. Early collaborative filtering methods primarily performed recommendation based on the similarity between users or items; for example, neighborhood-based collaborative filtering sought similar users or items through statistical measures such as cosine similarity and Pearson correlation coefficients, but

such methods are susceptible to inaccurate similarity estimation in data-sparse scenarios (Hu et al. 2008). Subsequently, matrix factorization methods mapped users and items into a low-dimensional latent space and modeled user preferences through latent vector interactions, becoming classic approaches in recommender systems. To enhance the nonlinear expressive capability of models, NCF replaced the traditional inner product function with neural networks to learn more complex user–item matching relationships (He et al. 2017; Wang et al. 2023).

In recent years, GNNs have been widely applied to recommendation tasks. NGCF modeled user–item interactions as a bipartite graph and propagated high-order collaborative signals through graph convolution to enhance user and item representations (Wang et al. 2019). LightGCN further pointed out that complex feature transformations and nonlinear activations are unnecessary in recommendation tasks, and therefore retained only the neighbor aggregation operation, effectively improving model efficiency and recommendation performance (He et al. 2020). In addition, several studies have further improved graph recommendation models from perspectives such as interest-aware message passing, low-pass collaborative filtering, and deep graph convolution, so as to enhance user–item representation learning (Liu et al. 2021; Yu and Qin 2020; Chen et al. 2020, 2021a). Building on this, DGCF modeled different types of collaborative relationships from the perspective of user intent disentanglement, while PinSage designed graph sampling and aggregation strategies for large-scale recommendation scenarios, improving the scalability of graph recommendation methods in practical systems (Wang et al. 2020; Ying et al. 2018).More recent graph-based recommender systems further explore advanced neural architectures, such as sheaf neural networks, to enhance representation learning on graph-structured recommendation data (Purificato et al. 2025).

With the development of self-supervised learning, graph contrastive learning has gradually become an important research direction in the recommendation domain (Ren et al. 2026). SGL constructed contrastive views through random graph augmentation to improve recommendation representation quality (Wu et al. 2021); LightGCL adopted a simple yet effective graph contrastive learning mechanism to reduce augmentation complexity (Cai et al. 2023); SimGCL and XSimGCL further demonstrated that simple perturbations or minimal augmentation can also effectively improve recommendation representations (Yu et al. 2022, 2023); DirectAU optimized the recommendation embedding distribution from the perspective of representation alignment and uniformity (Wu et al. 2022b). Recent studies have further extended the application of graph contrastive learning in recommendation tasks from directions such as adaptive graph contrastive learning, temporal graph contrastive learning, neighbor-layer contrastive learning, heterogeneous graph contrastive learning, and multi-objective contrastive optimization (Jiang et al. 2023; Peng et al. 2025; Xu et al. 2025; Sang et al. 2025a; Zhang et al. 2025b). Recent work published in the International Journal of Machine Learning and Cybernetics also explores dual contrastive learning for knowledge-aware recommendation (Wei et al. 2025) and intent-enhanced multi-side contrastive learning that addresses the under-representation of niche interests (Zeng et al. 2026). Although these approaches draw on auxiliary knowledge graphs

rather than explicitly estimating the reliability of observed social edges, they emphasize the complementary roles of contrastive alignment and preference diversity. These methods have effectively advanced the performance of general recommendation, but they primarily focus on user–item interaction graphs and give insufficient consideration to noisy social relations and item bias problems in social recommendation scenarios.

2.2 Social Recommendation

Social recommendation provides additional information for user preference learning by incorporating trust, follow, or friendship relationships among users, and is capable of alleviating data sparsity and cold-start problems in traditional collaborative filtering (Chen et al. 2019a; Ma et al. 2011). Early methods typically incorporated social relations as regularization constraints into matrix factorization models, so that users with social connections maintain similar representations in the latent space. For example, SoReg constrained user latent vectors using social regularization terms (Ma et al. 2011), TrustMF modeled the influence relationships among users from the perspective of trust propagation (Yang et al. 2017), and TrustSVD further combined rating behaviors and trust information to improve recommendation performance in sparse scenarios (Guo et al. 2015).

With the development of GNNs, researchers began to explicitly model social influence propagation using graph structures. GraphRec incorporated user interaction behaviors and social relations into a unified GNN framework to learn user preference representations (Fan et al. 2019); DANSER separately modeled user–item interactions and user–user social influence through a dual graph attention mechanism (Chen et al. 2019b); KCGN introduced a knowledge-aware graph convolution mechanism that combined external knowledge with social relations to enrich user and item representations (Chen et al. 2021c). The aforementioned methods demonstrated the effectiveness of social information for recommendation tasks, but most of them assumed by default that social connections are beneficial and lacked reliability assessment of low-quality social edges.

Related social recommendation studies in this journal make the reliability challenge particularly relevant. Gu et al. (2026) reconstruct social connectivity using user similarities extracted from the social and interaction graphs, whereas Li et al. (2026a) explicitly consider low-homophily social links and rating bias through a noise-tolerant graph neural architecture. These contributions reinforce the importance of distinguishing observed social ties from preference-consistent neighbors. Our model differs in using a residual similarity gate to retain each observed edge with a positive weight and a separate norm-bounded item-refinement branch; these design distinctions do not imply empirically superior rankings without direct controlled comparisons.

In recent years, research on social relation noise has attracted increasing attention. SEPT enhanced user representations through self-supervised utilization of tripartite relations to alleviate interaction sparsity (Yu et al. 2021a); MHCN captured high-order social relations of different semantics from a multi-channel hypergraph structure to improve user representation capability (Yu et al. 2021b). Furthermore, diffusion

generative models, heterogeneous hypergraph neural networks, and dual graph denoising models have also been applied to social recommendation scenarios to enhance the model's ability to handle complex social relations and noisy structures (Liu et al. 2025; Khan et al. 2025; Li and Yang 2025). However, existing methods primarily focus on the enhanced utilization of social information, and adaptive modeling of whether social edges are reliable remains insufficient; moreover, bias factors such as item popularity that interfere with recommendation representations are rarely considered. To address these challenges, our work investigates lightweight, confidence-weighted propagation and conservative item refinement. Among more recent alternatives, RecDiff uses task-aware multi-step diffusion to denoise social representations (Li et al. 2024), whereas RGCML uses opinion-dynamics-inspired social modeling, semantic fusion, and global–local contrastive objectives (Xiong et al. 2025). These methods tackle social reliability through different mechanisms. In contrast, our design couples direct reliability weighting of observed social edges with an explicit upper bound on item-representation correction; Section 3.6 explains these design-level distinctions without treating them as unmeasured empirical wins.

2.3 Model Optimization in Recommendation

The optimization objectives of recommendation models generally include two categories: rating prediction and Top-N ranking recommendation. Early methods mostly adopted pointwise loss functions, treating user ratings or interactions with items as supervised signals for learning. Weighted matrix factorization methods alleviated the uncertainty of negative samples in implicit feedback by assigning different weights to observed and unobserved samples (Hu et al. 2008). Subsequently, the BPR loss was widely adopted for ranking recommendation tasks; by constructing user–positive sample–negative sample triplets, it enabled models to directly learn that positive samples should be ranked higher than negative samples (Rendle et al. 2009).

However, the performance of sampling-based optimization methods is highly dependent on negative sample selection strategies. Random negative sampling is simple and efficient but tends to select uninformative negative samples; adaptive negative sampling and hard negative mining can enhance training signals but may also introduce false negatives, affecting model stability (Huang et al. 2021). To reduce sampling bias, some non-sampling methods directly optimize over the entire interaction matrix, but such methods also require careful design of sample weights; otherwise, models are easily dominated by high-frequency interactions and popular items.

In addition to sampling strategies, data bias is also an important factor affecting the optimization of recommendation models. User interactions are typically influenced by factors such as exposure opportunities, item popularity, and historical rating distributions, and models are prone to learning biased signals unrelated to genuine interests. In recent years, debiasing recommendation and counterfactual learning methods have attempted to identify and suppress the influence of bias factors from a causal effect perspective (Yang et al. 2025b). For example, AutoDebias improved recommendation robustness by automatically learning debiasing strategies (Chen et al. 2021b); debiased contrastive learning methods improved recommendation performance by reducing the interference of biased signals on representation learning (Yang et al. 2023); knowledge

graph debiasing methods enhanced recommendation credibility from the perspective of external semantic relations (Li et al. 2026b); and counterfactual augmentation and counterfactual effect estimation methods alleviated recommendation bias from the perspective of counterfactual samples or causal effects (Huang et al. 2025b; Wang et al. 2025; Tang et al. 2025; Yu et al. 2025; Sang et al. 2025b). Drawing on the idea of counterfactual debiasing, this work constructs item-side counterfactual directions during item representation learning and suppresses potential bias components through a weak residual strategy, thereby enhancing the model's ability to characterize genuine user preferences.

Beyond debiasing optimization, the robustness and long-term benefit optimization of recommender systems have also attracted increasing attention. Related studies have analyzed the stability and security of recommendation models in complex environments from perspectives such as robust recommendation surveys, attack detection and defense, and controllable exploration–exploitation mechanisms(Zhang et al. 2025a; Feng et al. 2026; Ning et al. 2026; Guo et al. 2026). These studies indicate that recommendation models need not only to pursue short-term accuracy but also to improve resistance to data noise, bias factors, and potential attacks.

3 Methodology

This section presents the RIBGCL model proposed in this paper. The model is designed for social recommendation tasks; it takes a user–item interaction graph and a user–user social graph as input, learns interest-domain and social-domain representations through graph adaptive propagation, further incorporates social confidence gating to alleviate the influence of noisy social relations, and designs a counterfactual item debiasing module to suppress potential bias directions in item representations. Finally, the model is jointly optimized through self-adaptive weighted supervised learning and augmentation-free dual contrastive learning.

As illustrated in Figure 3, the RIBGCL pipeline consists of four main modules: a data input module, a graph adaptive propagation module, a contrastive learning module, and a model optimization module.

The data input module provides the user–item interest graph and the user–user social graph. The graph adaptive propagation module is responsible for representation learning and further incorporates two key mechanisms: social confidence gating and counterfactual item debiasing. The contrastive learning module constructs multi-view self-supervised signals in an augmentation-free manner to enhance the quality of user and item representations. The model optimization module integrates the self-adaptive weighted recommendation loss, social supervision loss, contrastive learning loss, and counterfactual regularization term to achieve joint training of the model.

3.1 Problem Formulation

For clarity, the main notations used in this paper are summarized in Table 1.

Let $\mathcal{U}$ denote the user set and $\mathcal{I}$ denote the item set, where $|\mathcal{U}|$ and $|\mathcal{I}|$ are the numbers of users and items, respectively. The user–item interaction matrix is denoted as:

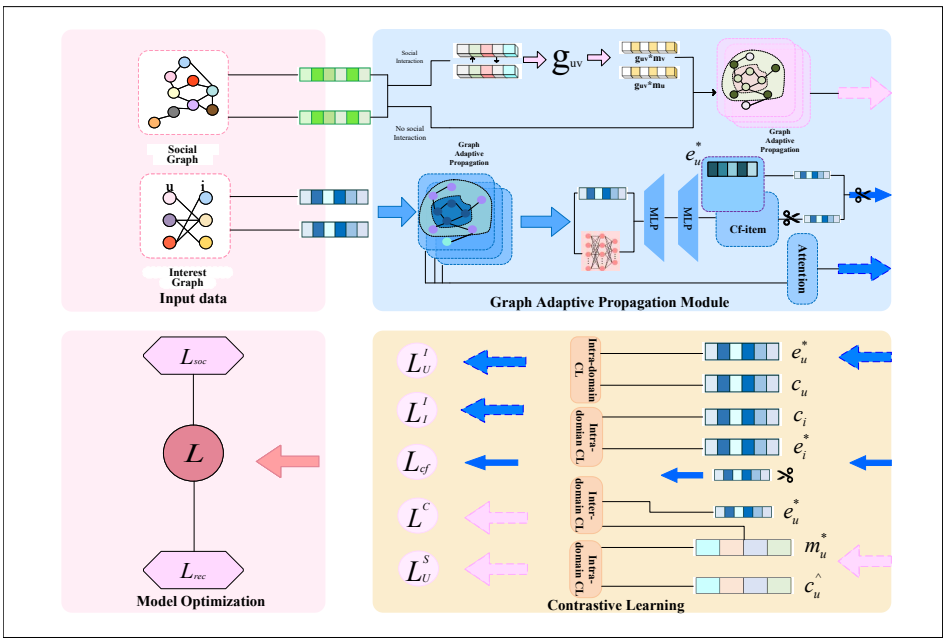


Fig. 3 The overall framework of RIBGCL.

Table 1 Main notations used in this paper.

Notation	Description
$\mathcal{U}$	User set
$\mathcal{I}$	Item set
$\mathbf{R}$	User–item interaction matrix
r_{ui}	Interaction indicator between user u and item i
$\mathbf{S}$	User–user social relation matrix
s_{uv}	Social relation indicator between user u and user v
$\mathcal{G}_I$	User–item interest graph
$\mathcal{G}_S$	User–user social graph
$\mathbf{e}_u$	Interest-domain representation of user u
$\mathbf{e}_i$	Interest-domain representation of item i
$\widetilde{\mathbf{e}}_i$	Debiased representation of item i
$\mathbf{m}_u$	Social-domain representation of user u
$\mathbf{c}_i^{cf}$	Counterfactual item vector of item i
g_{uv}	Confidence score of social edge (u, v)
α_i	Debiasing coefficient of item i
λ_g	Residual coefficient of social confidence gating
λ_{scale}	Scaling coefficient of counterfactual debiasing
$\hat{y}_{ui}$	Predicted preference score of user u for item i

$$\mathbf{R} \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{I}|}. \tag{1}$$

where $r_{ui} = 1$ indicates that user u has interacted with item i, and $r_{ui} = 0$ otherwise. Based on the interaction matrix, the user–item interest graph is constructed as:

$$\mathcal{G}_I = (\mathcal{U} \cup \mathcal{I}, \mathcal{E}_I)\,. \tag{2}$$

where $\mathcal{E}_I$ denotes the interaction edges between users and items. The interest graph is used to model users' historical behaviors and item-level collaborative relations.

Similarly, the social relations among users are represented by the social matrix:

$$\mathbf{S} \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{U}|} \tag{3}$$

where $S_{uv} = 1$ indicates that there exists a social relation between user u and user v, and $S_{uv} = 0$ otherwise. Based on the social matrix, the user–user social graph is constructed as:

$$\mathcal{G}_S = (\mathcal{U}, \mathcal{E}_S) \tag{4}$$

where $\mathcal{E}_S$ denotes the social edges among users. The social graph is used to model social influence and potential interest similarity among users.

The objective of this work is to learn user representations and item representations, and then predict the preference score of user u for candidate item i. In RIBGCL, the final recommendation prediction is performed by using the interest-domain user representation and the counterfactually debiased item representation. The prediction function is defined as:

$$y_{ui} = \mathbf{e}_u^T \widetilde{\mathbf{e}}_i \tag{5}$$

where $\widetilde{\mathbf{e}}_i$ denotes the counterfactually debiased item representation, and $\mathbf{e}_i$ denotes the debiased item representation. For each user, the model ranks all unobserved items in descending order according to the predicted scores and generates a Top-K recommendation list.

3.2 Data Input Module

3.2.1 Interest Graph and Social Graph Input

The input of RIBGCL consists of two graph structures: the user–item interest graph $\mathcal{G}_I$ and the user–user social graph $\mathcal{G}_S$.

The user–item interest graph $\mathcal{G}_I$ is constructed from the interaction matrix $\mathbf{R}$, and is used to describe users' historical behavioral preferences. It is a bipartite graph, in which nodes include both user nodes and item nodes, and each edge represents an observed interaction between a user and an item. By propagating information over the interest graph, the model is able to capture direct user–item interactions as well as high-order collaborative signals.

The user–user social graph $\mathcal{G}_S$ is constructed from the social matrix $\mathbf{S}$, and is used to describe trust, following, or friendship relations among users. All nodes in this graph are users, and each edge indicates the existence of a social connection between two users. By propagating information over the social graph, the model is able to exploit social neighbors to supplement user preference information, thereby alleviating data sparsity and cold-start issues.

It should be noted that low-quality or noisy social edges may exist in real-world social graphs. Therefore, RIBGCL did not directly regard the original social graph as

the only propagation basis. Instead, a social relation confidence-gating mechanism was further introduced in the graph adaptive propagation module to dynamically model the reliability of social edges.

3.2.2 Initial Embedding Representations

The model initializes trainable embedding matrices for users and items. The user initial embedding matrix is denoted as:

$$\mathbf{P} \in \mathbb{R}^{|\mathcal{U}| \times d} \tag{6}$$

The item initial embedding matrix is denoted as:

$$\mathbf{Q} \in \mathbb{R}^{|\mathcal{I}| \times d} \tag{7}$$

where d is the embedding dimension. For user u and item i, their initial representations are denoted as $\mathbf{p}_u$ and $\mathbf{q}_i$, respectively.

In the implementation, the user initial embedding corresponds to `self.user_embeddings`, and the item initial embedding corresponds to `self.item_embeddings`. These two embedding matrices serve as the basic inputs for interest graph propagation and social graph propagation, and are jointly updated with other model parameters during training.

3.3 Graph Adaptive Propagation Module

RIBGCL first performs graph propagation on the interest graph and the social graph to learn structured representations of users and items. For the shared propagation and training backbone, we adopt EGCL's degree-adaptive propagation, layer readout, and augmentation-free contrastive objectives (Wu et al. 2026); our added mechanisms are the social reliability gate and bounded item refinement.

3.3.1 Interest Graph Propagation

In the interest graph, users and items form a bipartite graph. The model first constructs the symmetric normalized adjacency matrix based on the user–item interaction matrix:

$$\mathbf{A}_I^* = \mathbf{D}_I^{-\frac{1}{2}} \mathbf{A}_I \mathbf{D}_I^{-\frac{1}{2}} \tag{8}$$

where $\mathbf{A}_I$ denotes the adjacency matrix of the interest graph, and $\mathbf{D}_I$ denotes the corresponding degree matrix.

At the k-th propagation layer, the normalized adjacency matrix is used for information aggregation. Meanwhile, a propagation control factor is generated according to node degree, and a heat-kernel-style layer weight is used to regulate the propagation result. For node x, the propagation control factor is defined as:

$$t_x = \sigma(\log(d_x + \epsilon)) \tag{9}$$

where d_x denotes the degree of node x, and ϵ is a small constant. In the implementation, ϵ is set to $1e^{-7}$.

At the k-th layer, the propagation weight is defined as:

$$\phi_x^{(k)} = \exp(-t_x)\frac{t_x^k}{k!} \tag{10}$$

Therefore, the user and item representations at the k-th layer in the interest graph are scaled according to the corresponding propagation weights. This mechanism allows users and items with different degrees to receive different information propagation strengths, thus avoiding the use of an identical propagation rule for all nodes.

After multi-layer propagation, the model obtains user and item representations at different layers. Then, an attention readout mechanism is used to fuse multi-layer representations, producing the final interest-domain user representation $\mathbf{E}_U$ and interest-domain item representation $\mathbf{E}_I$.

3.3.2 Social Confidence Gating Module

Noisy relations usually exist in real-world social networks. Even if two users are socially connected, they do not necessarily share similar interests (Hu et al. 2025; Zhang et al. 2025a; Li and Yang 2025). If the original social graph is directly used for propagation, unreliable social information may be introduced into user representations. To address this issue, RIBGCL introduced a social relation confidence-gating module to dynamically estimate the reliability of social edges according to interest-domain user representations.

For a social edge (u, v) in the social graph, the model first retrieves the interest-domain representations $\mathbf{e}_u$ and $\mathbf{e}_v$ of the two users, and then calculates their cosine similarity. Subsequently, the raw gating value is obtained through a Sigmoid function:

$$g_{uv}^{raw} = \sigma(\eta \cdot \cos(\mathbf{e}_u, \mathbf{e}_v)) \tag{11}$$

where η is the gate scaling coefficient and corresponds to `social_gate_scale` in the implementation.

To prevent useful social relations from being excessively weakened by overly strong gating, a residual confidence design was adopted:

$$g_{uv} = 1 - \lambda_g + \lambda_g g_{uv}^{raw} \tag{12}$$

where λ_g corresponds to `social_gate_lambda` in the implementation. This design ensures that social edges are not directly removed, but are weakly reweighted according to their confidence scores.

After the confidence score of each social edge is obtained, a weighted social graph is constructed. For an undirected social edge (u, v), the model sets:

$$\widetilde{S}_{uv} = g_{uv}, \quad \widetilde{S}_{vu} = g_{uv} \tag{13}$$

Then, the weighted social graph is symmetrically normalized as:

$$S^* = \widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{S}}\widetilde{\mathbf{D}}^{-\frac{1}{2}} \tag{14}$$

where $\widetilde{\mathbf{S}}$ denotes the weighted social adjacency matrix, and $\widetilde{\mathbf{D}}$ denotes its degree matrix. When social gating is enabled, the normalized weighted social graph is used as the propagation matrix in the social graph propagation stage.

The social relation confidence-gating module was designed to reduce the propagation strength of low-confidence social edges while preserving the overall structure of the social graph, with the aim of improving robustness to noisy social relations.

3.3.3 Social Graph Adaptive Propagation

In the social graph propagation stage, the model learns social-domain user representations based on the normalized social graph. When the social relation confidence-gating module is enabled, the propagation matrix is the confidence-weighted social graph. Otherwise, the normalized adjacency matrix of the original social graph is used.

Similar to interest graph propagation, social graph propagation also includes degree-based adaptive propagation and attention readout. For user u, after multi-layer social propagation, the final social-domain representation is obtained as:

$$\mathbf{m}_u = \sum_{k=0}^{L-1} \beta_u^{(k)} \mathbf{m}_u^{(k)} \tag{15}$$

where $\mathbf{m}_u^{(k)}$ denotes the representation of user u at the k-th layer in the social graph, and $\beta_u^{(k)}$ denotes the corresponding layer attention weight. Finally, the social-domain user representation matrix is obtained as:

$$\mathbf{M} = \{\mathbf{m}_u \mid u \in \mathcal{U}\} \tag{16}$$

This representation incorporates information from the user's social neighborhood. When social gating is enabled, the learned social-domain representation is further regulated by the confidence scores of social edges.

3.3.4 Attention Readout Mechanism

Different propagation layers contain neighborhood information at different ranges. Shallow-layer representations focus more on direct neighbors, whereas deeper-layer representations contain higher-order collaborative signals. To adaptively select the importance of different layers, RIBGCL used an attention mechanism to fuse multi-layer representations.

For user u, the model concatenates the representation at each propagation layer with the initial embedding, and then calculates layer-level weights through an attention network. The final interest-domain user representation is obtained as:

$$\mathbf{e}_u = \sum_{k=0}^{L-1} \beta_u^{(k)} \mathbf{e}_u^{(k)} \tag{17}$$

where $\mathbf{e}_u^{(k)}$ denotes the interest-domain representation of user u at the k-th layer, and $\beta_u^{(k)}$ denotes the corresponding layer attention weight.

The same readout strategy is used on the item side, and the final interest-domain item representation is obtained as:

$$\mathbf{e}_i = \sum_{k=0}^{L-1} \gamma_i^{(k)} \mathbf{e}_i^{(k)} \tag{18}$$

where $\gamma_i^{(k)}$ denotes the layer attention weight of item i.

Through attention readout, the model was able to adaptively fuse multi-layer propagation information according to the structural characteristics of different users and items.

3.3.5 Counterfactual Item Debiasing Module

User–item interaction data usually contain popularity bias, exposure bias, and historical behavior bias (Chen et al. 2021b; Yang et al. 2023; Gao et al. 2023). If item representations obtained from interest graph propagation are directly used for recommendation, these bias signals may be further amplified. To alleviate this issue, RIBGCL adds a counterfactual-inspired item-refinement module (Huang et al. 2025b; Wang et al. 2025; Tang et al. 2025).

The auxiliary direction is constructed from the observed users interacting with each item and is used as a proxy for a potentially non-personalized component. Since only observational user–item interactions are available, this branch does not define a causal intervention or identify a counterfactual outcome. The similarity to the original item embedding controls the magnitude of a conservative residual correction.

First, the model constructs a user–item interaction matrix normalized by item columns. For item i, the number of its interacted users is defined as:

$$d_i = \sum_{u \in U} r_{ui} \tag{19}$$

The column-normalized interaction matrix is denoted as $\overline{\mathbf{R}}$, and its element is defined as:

$$\overline{r}_{ui} = \begin{cases} \frac{r_{ui}}{d_i}, & d_i > 0, \\ 0, & d_i = 0. \end{cases} \tag{20}$$

This normalization ensures that the counterfactual representation of each item is obtained by averaging its interacted user representations, thereby preventing popular items from producing excessively large representation magnitudes due to a large number of interacted users.

When generating the counterfactual item vector, the model aggregates user representations by using the transpose of the normalized interaction matrix:

$$\mathbf{C}_{cf}^{(0)} = \overline{\mathbf{R}}^{\top} \mathbf{E}_U^{cf} \tag{21}$$

where $\mathbf{E}_U^{cf}$ denotes the user representation fed into the counterfactual branch. When `cf_stop_gradient` is enabled, the user representation is first detached from the gradient computation to prevent the counterfactual branch from interfering with the learning of the main user representations.

The initial counterfactual item representation $\mathbf{C}_{cf}^{(0)}$ is then fed into a two-layer MLP and L2-normalized to obtain the final counterfactual item representation:

$$\mathbf{C}_{cf} = \mathrm{L2Norm}(\mathrm{MLP}(\mathbf{C}_{cf}^{(0)})) \tag{22}$$

It should be emphasized that the final `cf_item_embeddings` are not simply averaged user-neighborhood representations. Instead, they are counterfactual item vectors obtained through neighborhood aggregation, a two-layer MLP, and L2 normalization.

3.3.6 Residual Weak Debiasing

After the counterfactual item representation is obtained, the model calculates the similarity between the original item representation $\mathbf{e}_i$ and the counterfactual item vector $\mathbf{c}_i^{cf}$, and obtains the debiasing coefficient through a Sigmoid function. To avoid over-removing useful collaborative signals, the implementation further scales the debiasing coefficient by `cf_scale`:

$$\alpha_i = \lambda_{scale} \cdot \sigma(\cos(\mathbf{e}_i, \mathbf{c}_i^{cf})) \tag{23}$$

where λ_{scale} corresponds to `cf_scale`. The nominal implementation default is 0.05; the validation-selected value in the reported sensitivity analysis is 0.10. These should not be conflated.

Finally, the debiased item representation is obtained as:

$$\widetilde{\mathbf{e}}_i = \mathbf{e}_i - \alpha_i \mathbf{c}_i^{cf} \tag{24}$$

This weak debiasing strategy was used to weaken the potential bias direction in item representations while retaining the original collaborative information. The counterfactual vector $\mathbf{c}_i^{cf}$ is treated as a potential bias direction because it is generated from the interacted user neighborhood of item i. Such neighborhood aggregation captures common behavioral and exposure patterns associated with the item, which may include popularity-related or non-personalized signals, especially for popular items. Therefore, $\mathbf{c}_i^{cf}$ provides an approximate direction for item-side bias rather than a complete description of the item representation.

To avoid removing useful collaborative filtering information, RIBGCL adopts a weak residual strategy. Since $\mathbf{c}_i^{cf}$ is L2-normalized and the debiasing coefficient satisfies

$$0 < \alpha_i < \lambda_{scale}, \tag{25}$$

the representation perturbation introduced by the debiasing operation is bounded by:

$$\|\widetilde{\mathbf{e}}_i - \mathbf{e}_i\|_2 = \alpha_i \left\|\mathbf{c}_i^{cf}\right\|_2 \leq \lambda_{scale}. \tag{26}$$

Thus, the refinement is norm-bounded by λ_{scale}, which limits the modification of the primary collaborative embedding. This bound alone does not establish that popularity or exposure bias has been causally removed. The effectiveness of this strategy relies on the assumption that the interacted user neighborhood contains bias-related common patterns. When this assumption is weak, for example when the item has very few interactions or the estimated counterfactual vector is highly correlated with useful collaborative signals, the small coefficient and regularization help reduce the risk of over-debiasing.

If the counterfactual module is disabled, the model directly uses the original interest-domain item representation as the final item representation.

3.4 Contrastive Learning Module

After the graph adaptive propagation module, the model obtains the interest-domain user representation $\mathbf{E}_U$, the debiased item representation $\widetilde{\mathbf{E}}_I$, and the social-domain user representation $\mathbf{M}$. Based on these representations, the contrastive learning module constructs augmentation-free dual contrastive learning objectives, including intra-interest-domain contrast, intra-social-domain contrast, and cross-domain contrast between the interest domain and the social domain.

3.4.1 Recommendation Prediction

After training, the model saves the interest-domain user representation as the final user representation and the debiased item representation as the final item representation:

$$\mathbf{E}_U^{final} = \mathbf{E}_U \tag{27}$$

$$\mathbf{E}_I^{final} = \widetilde{\mathbf{E}}_I \tag{28}$$

For target user u, the model calculates the predicted scores between the user and all candidate items as:

$$\hat{\mathbf{Y}}_u = \mathbf{e}_u(\widetilde{\mathbf{E}}_I)^T \tag{29}$$

Then, the candidate items are sorted in descending order according to the predicted scores, and the Top-K recommendation list is generated.

It should be noted that the social-domain user representation was mainly used during training to assist user representation learning through the social supervision loss and the inter-domain contrastive loss. During final prediction, the model uses the interest-domain user representation and the debiased item representation for scoring.

3.4.2 Intra-Domain Contrastive Learning

Graph contrastive learning has been widely used to enhance recommendation representations, since it can improve the quality of user and item embeddings through self-supervised signals (Cai et al. 2023; Wu et al. 2021; Yu et al. 2022, 2023). Given two groups of representations $\mathbf{X}$ and $\mathbf{Y}$, the model adopts the InfoNCE loss for contrastive learning:

$$\mathcal{L}_{NCE}(\mathbf{X}, \mathbf{Y}) = -\sum_n \log \frac{\exp(\text{sim}(\mathbf{x}_n, \mathbf{y}_n)/\tau)}{\sum_j \exp(\text{sim}(\mathbf{x}_n, \mathbf{y}_j)/\tau)} \tag{30}$$

where $\text{sim}(\cdot, \cdot)$ denotes the inner product similarity after normalization, and τ is the temperature coefficient.

The interest-domain user contrast is used to constrain the consistency between user representations and their interacted item-neighborhood representations. In RIBGCL, the item-neighborhood representation of users is aggregated from the debiased item representations:

$$\mathbf{C}_U^I = \mathbf{R}\widetilde{\mathbf{E}}_I \tag{31}$$

Therefore, the interest-domain user contrastive loss is defined as:

$$\mathcal{L}_U^I = \mathcal{L}_{NCE}\left(\mathbf{E}_U, \mathbf{C}_U^I\right) \tag{32}$$

The interest-domain item contrast is used to constrain the consistency between debiased representations and their interacted user-neighborhood representations. The user-neighborhood representation of items is defined as:

$$\mathbf{C}_I^I = \mathbf{R}^\top \mathbf{E}_U \tag{33}$$

The corresponding interest-domain item contrastive loss is defined as:

$$\mathcal{L}_I^I = \mathcal{L}_{NCE}\left(\widetilde{\mathbf{E}}_I, \mathbf{C}_I^I\right) \tag{34}$$

The social-domain user contrast is used to constrain the consistency between social-domain user representations and their social-neighborhood representations. The social-neighborhood representation is defined as:

$$\mathbf{C}_U^S = \overline{\mathbf{S}}\mathbf{M} \tag{35}$$

The corresponding social-domain user contrastive loss is defined as:

$$\mathcal{L}_U^S = \mathcal{L}_{NCE}\left(\mathbf{M}, \mathbf{C}_U^S\right) \tag{36}$$

It should be noted that $\mathbf{S}$ in social-domain contrastive learning comes from the column-normalized original social matrix, rather than the weighted social propagation matrix generated by the social relation confidence-gating module. The social gating mechanism mainly affects the social-domain user representation $\mathbf{M}$ obtained from social graph propagation.

3.4.3 Inter-Domain Contrastive Learning

The interest graph and the social graph describe users' historical behavioral preferences and social influence, respectively. To align user representations from the two domains, the model treats the interest-domain representation and social-domain representation of the same user as a positive pair, and constructs the inter-domain contrastive loss as:

$$\mathcal{L}_C = \mathcal{L}_{NCE}(\mathbf{E}_U, \mathbf{M}) \tag{37}$$

This loss encourages the representations of the same user in the interest domain and the social domain to be closer, while enhancing the distinction among different users.

3.4.4 Total Contrastive Learning Loss

The three intra-domain contrastive losses jointly constitute the intra-domain contrastive learning objective:

$$\mathcal{L}_{intra} = \mathcal{L}_U^I + \mathcal{L}_I^I + \mathcal{L}_U^S \tag{38}$$

The inter-domain contrastive learning objective is defined as:

$$\mathcal{L}_{inter} = \mathcal{L}_C \tag{39}$$

Therefore, the total contrastive learning loss is defined as:

$$\mathcal{L}_{CL} = \lambda_{intra}\mathcal{L}_{intra} + \lambda_{inter}\mathcal{L}_{inter} \tag{40}$$

where λ_{intra} and λ_{inter} correspond to `intra_reg` and `inter_reg` in the implementation, respectively.

3.5 Model Optimization Module

This module is responsible for constructing the final training objective of RIBGCL and updating model parameters through joint optimization. The model optimization objective primarily consists of the self-adaptive weighted recommendation loss, the social supervision loss, the contrastive learning loss, and the counterfactual regularization term.

3.5.1 Self-Adaptive Weighted Supervised Learning

RIBGCL used a non-sampling supervised learning objective for recommendation optimization and social relation optimization. Different from negative-sampling-based methods, the non-sampling loss incorporates all unobserved samples into the optimization objective and enables efficient computation through matrix operations. Meanwhile, the model dynamically generates adaptive weights for observed positive samples to distinguish the contributions of different positive samples during training.

For a positive user–item pair (u, i), the model uses the interest-domain user representation $\mathbf{e}_u$ and the debiased item representation $\mathbf{e}_i$ to generate the positive sample weight. In the implementation, the two representations are first detached from gradient computation. Then, the user representation, the item representation, and their element-wise product are concatenated:

$$\mathbf{x}_{ui} = [\mathbf{e}_u \| \widetilde{\mathbf{e}}_i \| \mathbf{e}_u \odot \widetilde{\mathbf{e}}_i] \tag{41}$$

After that, a two-layer MLP is used to generate the recommendation positive sample weight:

$$w_{ui}^{rec} = \text{softplus}\left(\text{MLP}(\mathbf{x}_{ui})\right) + 8 \tag{42}$$

where the constant 8 is the base offset of the recommendation positive sample weight in the implementation. This weight is used to measure the importance of different user–item positive interaction samples during training.

For a positive user–user social pair (u, v), the model uses the social-domain user representations $\mathbf{m}_u$ and $\mathbf{m}_v$ to generate the social positive sample weight. Similarly, the model concatenates the two user representations and their element-wise product:

$$\mathbf{x}_{uv} = [\mathbf{m}_u \| \mathbf{m}_v \| \mathbf{m}_u \odot \mathbf{m}_v] \tag{43}$$

Then, a two-layer MLP is used to obtain the social positive sample weight:

$$w_{uv}^{soc} = \text{softplus}\left(\text{MLP}(\mathbf{x}_{uv})\right) + 10 \tag{44}$$

where the constant 10 is the base offset of the social positive sample weight in the implementation. This weight is used to dynamically distinguish the contributions of different social relations in supervised learning.

3.5.2 Recommendation Supervision Loss and Social Supervision Loss

The recommendation prediction score is defined as:

$$\hat{y}_{ui} = \mathbf{e}_u^\top \widetilde{\mathbf{e}}_i \tag{45}$$

The social prediction score is defined as:

$$s_{uv} = \mathbf{m}_u^T \mathbf{m}_v \tag{46}$$

The model constructs the recommendation supervision loss and the social supervision loss, respectively, and controls the importance of the social supervision loss through parameter β. The basic supervised loss of the embedding branch is defined as:

$$\mathcal{L}_{embed} = \mathcal{L}_{rec}^{embed} + \beta \mathcal{L}_{soc}^{embed} \tag{47}$$

where $\mathcal{L}_{rec}^{embed}$ is calculated using the interest-domain user representations and the debiased item representations, and $\mathcal{L}_{soc}^{embed}$ is calculated using the social-domain user representations.

Meanwhile, an additional weight-branch loss is constructed in the implementation to update the recommendation weight generation network and the social weight generation network:

$$\mathcal{L}_{weight} = \mathcal{L}_{rec}^{weight} + \beta \mathcal{L}_{soc}^{weight} \tag{48}$$

During training, two optimizers were used to update the weight branch and the embedding branch separately, so as to reduce the mutual interference between the weight generation networks and the backbone representation learning process.

3.5.3 Counterfactual Regularization Term

To prevent the counterfactual debiasing module from producing an overly strong debiasing effect, a squared mean regularization term is imposed on the debiasing coefficients:

$$\mathcal{L}_\alpha = \lambda_{cf} \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \alpha_i^2 \tag{49}$$

Meanwhile, L2 regularization is applied to the parameters of the counterfactual MLP. Together, these two parts form the counterfactual regularization term:

$$\mathcal{L}_{cf} = \mathcal{L}_\alpha + \mathcal{L}_{cf-param} \tag{50}$$

where λ_{cf} corresponds to `cf_reg` in the implementation. This regularization term is included separately in the embedding-branch total objective to constrain the correction strength and reduce the risk of excessive representation changes.

Algorithm 1 Two-branch optimization and inference for RIBGCL

Require: Interaction graph G_I, social graph G_S, interaction matrix R, hyper-parameters λ_{intra}, λ_{inter}, β, λ_{cf}, learning rates η_w, η_e

Ensure: User representations E_U and refined item representations $\widetilde{E}_I$

Initialize embedding/backbone parameters Θ_e and sample-weight parameters Θ_w
while validation-based stopping criterion is not satisfied **do**
    Obtain E_U, E_I by interest-graph propagation on G_I
    Construct confidence weights g_{uv} using Eqs. (11)–(12); obtain M from weighted social propagation
    Build C_{cf} and $\widetilde{E}_I$ via Eqs. (19)–(24)
    Evaluate recommendation and social weight objectives $\mathcal{L}_{weight}$ from Eq. (52)
    Update only Θ_w using $\nabla_{\Theta_w}\mathcal{L}_{weight}$ and learning rate η_w
    Evaluate $\mathcal{L}_{embed}$, $\mathcal{L}_{cf}$, and contrastive terms with current sample weights
    $\mathcal{L}_{total} \leftarrow \mathcal{L}_{embed} + \mathcal{L}_{cf} + \lambda_{\text{intra}}(\mathcal{L}_U^I + \mathcal{L}_I^I + \mathcal{L}_U^S) + \lambda_{\text{inter}}\mathcal{L}^C$
    Update only Θ_e using $\nabla_{\Theta_e}\mathcal{L}_{total}$ and learning rate η_e
end while
Recompute representations using the selected checkpoint
return $E_U, \widetilde{E}_I$ for scores $\widehat{Y} = E_U\widetilde{E}_I^\top$

3.5.4 Final Optimization Objective

The final optimization objective of the embedding branch is defined as:

$$\begin{aligned}\mathcal{L}_{SSL}^{SL} = \mathcal{L}_{embed} + \mathcal{L}_{cf} \\ + \lambda_{intra}(\mathcal{L}_U^I + \mathcal{L}_I^I + \mathcal{L}_U^S) \\ + \lambda_{inter}\mathcal{L}^C\end{aligned} \tag{51}$$

Here $\mathcal{L}_{embed}$ denotes the recommendation and social supervised embedding terms from Eq. (47). The auxiliary correction penalty $\mathcal{L}_{cf}$ is added separately, avoiding an ambiguous double count.

The optimization objective of the weight branch is defined as:

$$\mathcal{L}_{weight} = \mathcal{L}_{rec}^{weight} + \beta\mathcal{L}_{soc}^{weight} \tag{52}$$

In the training description, the weight branch is updated first in each epoch, followed by the embedding branch. The weight branch was used to learn recommendation positive sample weights and social positive sample weights. The embedding branch was used to update user embeddings, item embeddings, attention parameters, social propagation parameters, and the parameters of the counterfactual debiasing module.

3.5.5 Model Training Procedure

The final training objective of RIBGCL is jointly constituted by the self-adaptive weighted supervision loss, the augmentation-free dual contrastive learning loss, and the counterfactual regularization term. Two optimizers are employed in the implementation to update the weight branch parameters and the embedding branch parameters separately.

3.6 Analytical Properties and Method-Level Comparison

In addition to the ranking results reported below, the two additional operators in RIBGCL admit several verifiable properties. These properties explain *when and how* the proposed formulation can be preferable to unregulated propagation or unrestricted representation correction; they do not constitute a proof of higher ranking accuracy than independently trained models.

Property 1: monotone social confidence with a residual floor.

For any observed social edge (u, v), let $s_{uv} = \cos(\mathbf{e}_u, \mathbf{e}_v) \in [-1, 1]$. By Eqs. (11)–(12), if $\eta > 0$ and $0 \leq \lambda_g \leq 1$, its pre-normalization weight satisfies

$$\begin{aligned} 1 - \lambda_g \leq g_{uv} \leq 1, \\ \frac{\partial g_{uv}}{\partial s_{uv}} = \lambda_g \eta \sigma(\eta s_{uv}) \\ \cdot [1 - \sigma(\eta s_{uv})] \geq 0. \end{aligned} \tag{53}$$

Thus, higher interest consistency cannot receive a smaller *raw* gate weight, and, for $\lambda_g < 1$, the mechanism avoids complete removal of any observed edge. Compared with unweighted social propagation, it provides an explicit edge-level attenuation rule; compared with hard threshold pruning, it does not delete low-confidence edges. Importantly, symmetric degree normalization in Eq. (14) may change the relative contributions after aggregation, so the monotonicity assertion applies to g_{uv} before graph normalization, not to an end-to-end guarantee on user-embedding quality.

Property 2: bounded item correction and prediction change.

For every item i, Eqs. (22)–(24) give $\widetilde{\mathbf{e}}_i = \mathbf{e}_i - \alpha_i \mathbf{c}_i^{cf}$ with $\|\mathbf{c}_i^{cf}\|_2 \leq 1$ and $0 < \alpha_i < \lambda_{\text{scale}}$. Therefore,

$$\begin{aligned} \|\widetilde{\mathbf{e}}_i - \mathbf{e}_i\|_2 \leq \lambda_{\text{scale}}, \\ |\mathbf{e}_u^\top \widetilde{\mathbf{e}}_i - \mathbf{e}_u^\top \mathbf{e}_i| \leq \lambda_{\text{scale}} \|\mathbf{e}_u\|_2. \end{aligned} \tag{54}$$

The second inequality follows from Cauchy–Schwarz and gives an explicit limit on how much the item-refinement branch alone can alter a recommendation score when the user embedding is held fixed. For two items i, j, the relative score is changed by at most $2\lambda_{\text{scale}}\|\mathbf{e}_u\|_2$; consequently, a baseline ranking margin larger than this bound cannot be reversed by this correction alone. This is a stability property of the refinement operator, *not* a claim that the learned direction necessarily removes popularity bias or improves every ranking.

Property 3: complementary control with limited additional operators.

The social gate changes propagation weights on $\mathcal{G}_S$, while the item correction acts on representations derived from $\mathcal{G}_I$. The mechanisms therefore regulate different quantities, rather than merely applying two denoisers to the same embedding. Given interest-domain embeddings, the added gate can be evaluated in $O(|\mathcal{E}_S|d)$ operations; computing the neighborhood averages for item correction costs $O(|\mathcal{E}_I|d)$, followed by

an item-wise two-layer MLP of approximately $O(|\mathcal{I}|dh)$ for hidden width h. This characterizes the added operators only, not the overall training complexity, whose contrastive and supervised objectives remain substantial.

Table 2 Mechanism-level comparison with representative and recent methods. Entries describe published design choices, not results from a common experimental protocol.

Method	**Primary treatment of social information**	**Item-side bias control in stated design**	**Distinguishing perspective**
LightGCN (He et al. 2020)	Interaction-graph propagation; no dedicated social-reliability gate	No explicit bounded correction	Lightweight collaborative propagation without modeled social-edge reliability
EGCL (Wu et al. 2026)	Social-graph propagation with adaptive depth / readout	No explicit bounded item-refinement branch	Provides the shared graph-learning foundation, but not the two new controls
RecDiff (Li et al. 2024)	Task-aware, multi-step hidden-space diffusion denoising	Bounded per-item residual is not its central mechanism	Learned diffusion denoising versus direct edge-confidence weighting
RGCML (Xiong et al. 2025)	Opinion-dynamics social modeling, semantic fusion, global–local contrast	Bounded per-item residual is not its central mechanism	Multi-view preference modeling versus explicit two-level reliability / refinement control
RIBGCL	**Similarity-monotone residual edge gate**	**Explicit norm-bounded item correction**	**Simultaneous control at social-edge and item-embedding levels**

Table 2 highlights the narrower but verifiable advantage of RIBGCL: it specifies how social-edge influence is bounded away from hard removal and how strongly its item-refinement branch may change a prediction. RecDiff and RGCML offer different and potentially more expressive noise-handling capabilities; the analysis does *not* establish that RIBGCL is more accurate, more efficient overall, or more robust under arbitrary social perturbations. Resolving such questions requires retraining all methods with identical datasets, splits, candidate pools, and metrics, which is left outside the present comparison table.

4 Experiments

This section presents the experimental evaluation of the proposed RIBGCL framework. The experiments were conducted on three public social recommendation datasets and compared with representative recommendation methods. The evaluation aims to answer the following research questions:

- **RQ1:** How does RIBGCL perform compared with representative general recommendation and social recommendation methods?
- **RQ2:** How efficient is RIBGCL compared with existing methods?

- **RQ3:** How do the proposed social confidence gating module and counterfactual item debiasing module contribute to the final recommendation performance?

The following subsections introduce the experimental setup, overall performance comparison, training efficiency analysis, and ablation study.

4.1 Experimental Setup

4.1.1 Datasets

Experiments were conducted on three publicly available social recommendation datasets: Epinions (http://www.trustlet.org/epinions.html), FilmTrust (https://github.com/guoguibing/librec/tree/3.0.0/data/filmtrust), and Ciao (https://www.cse.msu.edu/~tangjili/trust.html). These datasets contain both user-item interaction records and user-user social relations, and have been widely used in social recommendation research.

Epinions and Ciao are product review datasets, in which users can rate items and establish trust relationships with other users. FilmTrust is a movie recommendation dataset containing user ratings on movies and trust links among users. Therefore, these datasets are suitable for evaluating whether a recommendation model can effectively utilize collaborative signals and social information simultaneously.

In data preprocessing, explicit ratings were converted into implicit feedback. Specifically, if a user had rated or reviewed an item, the corresponding user–item pair was treated as a positive interaction. Unobserved user–item pairs were not directly regarded as negative samples in the data construction stage. For social relationship data, user–user trust links were retained as edges in the social graph.

For each dataset, user–item interaction records were randomly split into training, validation, and test sets with a ratio of 7:1:2. The training set was used for model optimization, the validation set was used for hyper-parameter tuning and model selection, and the test set was used for final performance evaluation. For users with few interactions, the splitting procedure ensured that at least one interaction was retained in the training set whenever possible, so as to avoid making users completely invisible during training. All comparison methods adopted the same data splitting strategy to ensure a fair comparison.

The statistics of the three benchmark datasets are reported in Table 3.

Table 3 Statistics of the three benchmark datasets.

Dataset	#Users	#Items	#Inter.	#Social	Sparsity	R/U	R/I
Epinions	2,985	20,000	214,477	18,178	99.64%	71.85	10.72
FilmTrust	1,508	2,071	35,494	1,853	98.86%	23.54	17.14
Ciao	7,267	11,211	147,995	111,603	99.82%	20.37	13.20

As shown in Table 3, all three datasets are highly sparse, which is consistent with real-world recommendation scenarios. Among them, Ciao contains the largest number of users and social relations, while FilmTrust is smaller and relatively denser. These

differences make the datasets suitable for evaluating the robustness and generalization ability of RIBGCL under different data scales, sparsity levels, and social relation densities.

4.1.2 Evaluation Metrics

Four widely used ranking-oriented metrics were adopted to evaluate recommendation performance: Precision@N, Recall@N, MAP@N, and NDCG@N. In this paper, N was set to 10.

Precision@10 measures the proportion of relevant items among the Top-10 recommended items. Recall@10 measures the proportion of relevant test items that are successfully retrieved by the Top-10 recommendation list. MAP@10 evaluates the average precision of the ranked list and reflects the overall ranking quality. NDCG@10 further considers the positions of relevant items in the recommendation list, assigning higher scores when relevant items appear at higher ranks.

For all metrics, a larger value indicates better recommendation performance.

4.1.3 Baseline Methods

To evaluate the effectiveness of RIBGCL, several representative recommendation methods were selected as baselines. These methods can be divided into two categories: general recommendation methods and social recommendation methods.

The general recommendation baselines include:

- **BPRMF** (Rendle et al. 2009): A classical matrix factorization method optimized by Bayesian personalized ranking loss.
- **LightGCN** (He et al. 2020): A lightweight graph collaborative filtering method that removes feature transformation and nonlinear activation from graph convolution.
- **NGCF** (Wang et al. 2019): A graph neural collaborative filtering model that captures high-order collaborative signals through graph propagation.

The social recommendation baselines include:

- **EATNN** (Chen et al. 2019c): An adaptive-transfer neural recommendation model that learns user preferences across interaction and social domains.
- **EAGCN** (Wu et al. 2022a): An adaptive graph convolutional recommender that jointly exploits item interactions and social graph structure.
- **EGCL** (Wu et al. 2026): The direct base model of this work, which combines graph adaptive propagation, augmentation-free contrastive learning, and self-adaptive weighted supervised learning.

Compared with EGCL, the proposed RIBGCL further introduces a social confidence gating module and a counterfactual item debiasing module. These two modules are designed to alleviate noisy social propagation and item-side bias, respectively. Recent, particularly relevant social-noise baselines include RecDiff (CIKM 2024) (Li et al. 2024) and RGCML (AAAI 2025) (Xiong et al. 2025). Likewise, the related GNN–DSF and noise-tolerant GNN studies (Gu et al. 2026; Li et al. 2026a) are discussed as prior art rather than treated as tested baselines. The published scores of RecDiff and RGCML are not inserted into Table 4, because their published experiments use

different dataset preprocessing or evaluation protocols; an apples-to-apples comparison requires running their implementations under the same split, candidate pool, and Top-10 metrics.

4.1.4 Implementation Details

The embedding dimension was set to 64 for all methods unless otherwise specified. Model parameters were optimized using the Adam optimizer. The learning rate was selected from {0.001, 0.01, 0.1}. For fair comparison, all baseline methods were tuned under the same data splits and evaluation protocol.

For RIBGCL, the social gate scaling coefficient `social_gate_scale` was searched in {1.0, 2.0, 3.0, 5.0}, and the social gate residual coefficient `social_gate_lambda` was searched in {0.3, 0.5, 0.7, 1.0}. The counterfactual debiasing scaling coefficient `cf_scale` was searched in {0.01, 0.05, 0.1, 0.2}. The best hyper-parameters were selected according to the validation performance.

To verify the influence of key hyper-parameters on model performance, a sensitivity analysis was conducted for `social_gate_scale`, `social_gate_lambda`, and `cf_scale`. During the analysis, one hyper-parameter was varied each time, while the others were fixed to their optimal values. Precision@10 was used as the evaluation metric.

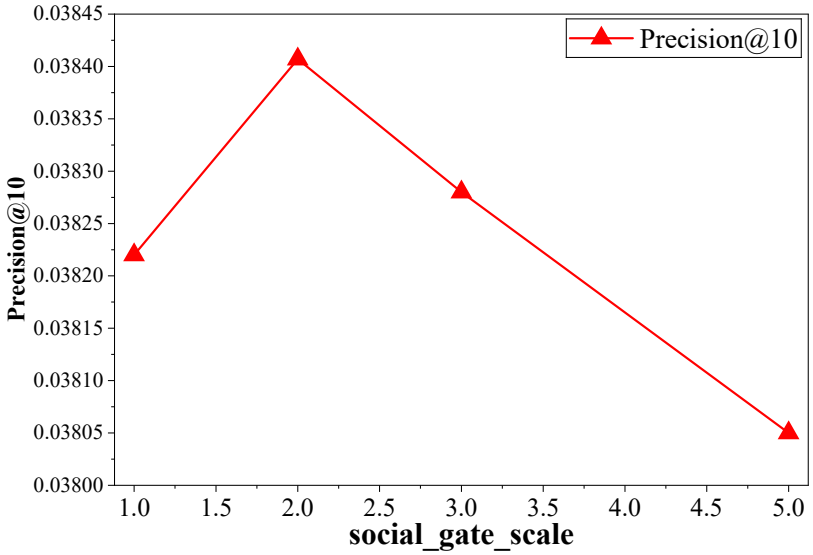


Fig. 4 Sensitivity of Precision@10 to the social-gate scaling coefficient (`social_gate_scale`).

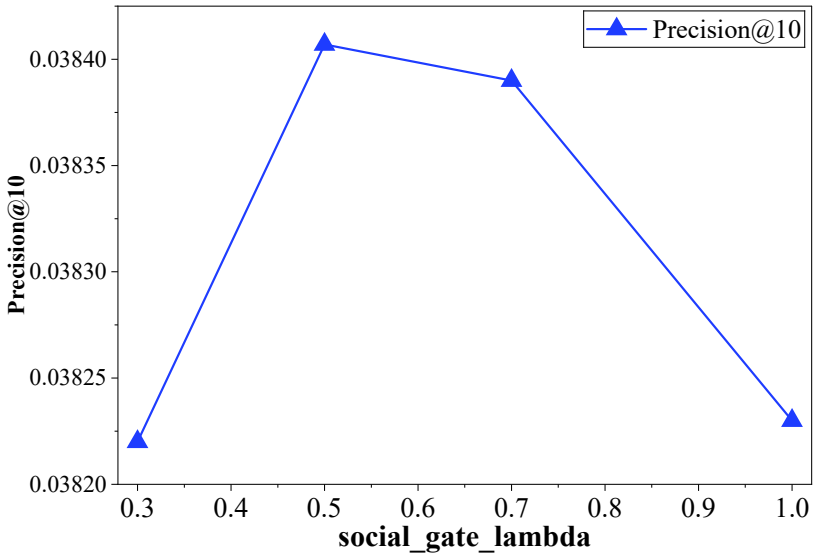


Fig. 5 Sensitivity of Precision@10 to the social-gate residual coefficient (`social_gate_lambda`).

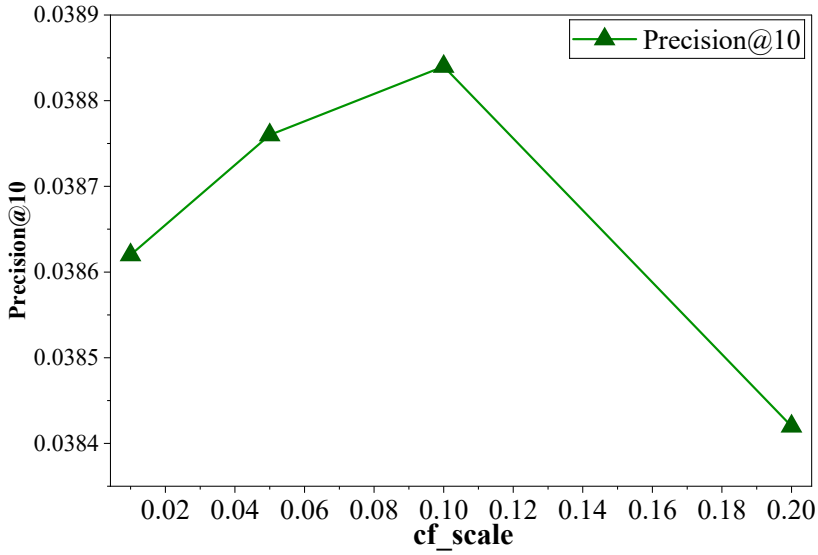


Fig. 6 Sensitivity of Precision@10 to the item-refinement scaling coefficient (`cf_scale`).

As shown in Figs. 4–6, the best performance was achieved when `social_gate_scale`, `social_gate_lambda`, and `cf_scale` were set to 2.0, 0.5, and 0.1, respectively.

For `social_gate_scale`, a small value made the confidence gating function less discriminative, while an overly large value made it too sensitive and weakened useful social relations. Therefore, 2.0 provided a better balance between suppressing noisy social edges and preserving reliable social information.

For `social_gate_lambda`, the optimal value was 0.5. A smaller value limited the effect of confidence-based reweighting, whereas a larger value excessively changed the original social graph structure. Thus, 0.5 allowed the model to filter unreliable social edges while retaining useful social collaborative signals. For `cf_scale`, the optimal value was 0.1. Smaller values produced insufficient counterfactual debiasing, while a larger value could cause over-debiasing and remove useful item collaborative information. These results indicate that moderate social denoising and debiasing strengths are more beneficial for RIBGCL.

All experiments were conducted on a workstation equipped with an Intel Core i7-12700H CPU and an NVIDIA RTX 6000 GPU. Each experiment was independently repeated five times, and the average results are reported. This repeated evaluation was used to reduce the influence of randomness caused by parameter initialization and data splitting.

4.2 Overall Performance Comparison

4.2.1 Recommendation Accuracy

Table 4 reports the recommendation performance of all methods on the three datasets. The best result in each column is highlighted in bold. The "Gain" row indicates the relative improvement of RIBGCL over the strongest listed baseline for each metric, calculated from the rounded entries.

As shown in Table 4, RIBGCL achieves the best performance on all four metrics across the three datasets *among the methods actually evaluated under this protocol*. These comparisons indicate improved ranking accuracy under the reported protocol. They do not independently measure popularity-bias removal, robustness to injected noisy edges, or the statistical significance of small improvements.

On the Ciao dataset, RIBGCL obtains the most significant improvement. Compared with EGCL, the strongest baseline on Ciao, the relative gains calculated from the rounded values in Table 4 are 6.89%, 10.70%, 12.14%, and 9.69% in Precision@10, Recall@10, MAP@10, and NDCG@10, respectively. Since Ciao is highly sparse and contains abundant social relations, this result indicates that RIBGCL can effectively exploit useful social information while suppressing the negative influence of noisy social edges.

On the Epinions dataset, RIBGCL also consistently outperforms all baseline methods. In particular, RIBGCL improves MAP@10 and NDCG@10 by 2.65% and 2.55%, respectively. This suggests that RIBGCL not only increases the probability of recommending relevant items, but also improves the ranking positions of relevant items in the recommendation list.

On the FilmTrust dataset, the improvement margin is relatively smaller. A possible reason is that FilmTrust is smaller and denser than the other two datasets, so baseline methods can already learn relatively sufficient collaborative signals. RIBGCL

Table 4 Overall performance comparison on three datasets.

Dataset	Method	Precision@10	Recall@10	MAP@10	NDCG@10
Ciao	BPRMF	0.0289	0.0630	0.0306	0.0544
	LightGCN	0.0325	0.0756	0.0369	0.0639
	NGCF	0.0299	0.0694	0.0335	0.0588
	EATNN	0.0323	0.0764	0.0341	0.0614
	EAGCN	0.0341	0.0828	0.0391	0.0674
	EGCL	0.0363	0.0841	0.0412	0.0712
	RIBGCL	**0.0388**	**0.0931**	**0.0462**	**0.0781**
	Gain	6.89%	10.70%	12.14%	9.69%
FilmTrust	BPRMF	0.3669	0.6765	0.4551	0.5581
	LightGCN	0.3610	0.6745	0.4581	0.5586
	NGCF	0.3672	0.6764	0.4632	0.5638
	EATNN	0.3678	0.6812	0.4403	0.5585
	EAGCN	0.3674	0.6809	0.4665	0.5613
	EGCL	0.3654	0.6781	0.4518	0.5545
	RIBGCL	**0.3698**	**0.6826**	**0.4691**	**0.5657**
	Gain	0.54%	0.21%	0.56%	0.34%
Epinions	BPRMF	0.0505	0.0376	0.0252	0.0598
	LightGCN	0.0583	0.0451	0.0311	0.0695
	NGCF	0.0512	0.0377	0.0255	0.0601
	EATNN	0.0655	0.0491	0.0355	0.0791
	EAGCN	0.0685	0.0518	0.0371	0.0819
	EGCL	0.0686	0.0511	0.0378	0.0825
	RIBGCL	**0.0699**	**0.0522**	**0.0388**	**0.0846**
	Gain	1.90%	0.77%	2.65%	2.55%

obtains the highest reported mean on all four metrics, but the gains over the strongest FilmTrust baselines are only 0.21%–0.56%. Standard deviations and paired significance tests are required before treating these small margins as conclusive.

Overall, the reported results support the effectiveness of the combined refinement framework across datasets with different interaction densities. Strong claims about generalization and noise robustness require direct perturbation tests and confidence intervals.

4.2.2 Visual Performance Analysis

To provide a more intuitive comparison, Figs. 7–9 present the performance comparison of different methods on the three datasets.

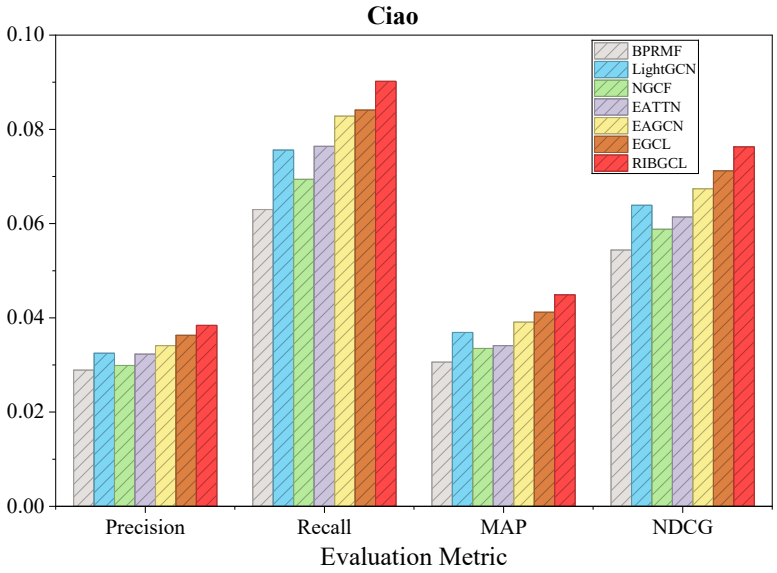


Fig. 7 Top-10 recommendation performance of the evaluated methods on Ciao.

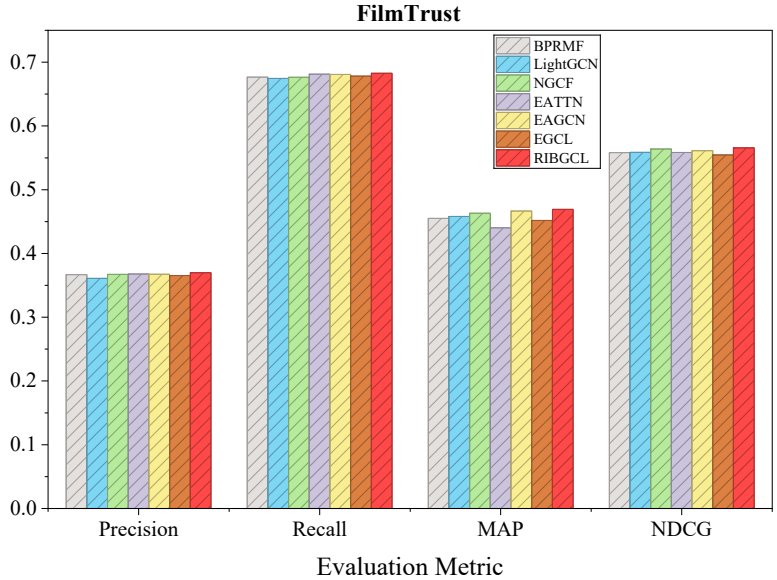


Fig. 8 Top-10 recommendation performance of the evaluated methods on FilmTrust.

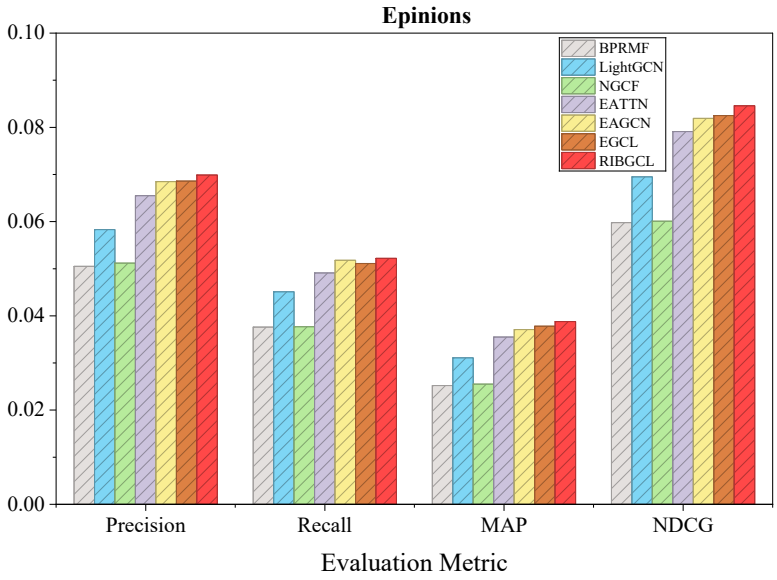


Fig. 9 Top-10 recommendation performance of the evaluated methods on Epinions.

As shown in Figs. 7–9, general recommendation methods such as BPRMF, LightGCN, and NGCF achieve stable performance on some datasets, but their overall performance is generally lower than that of social recommendation methods. This is because they mainly rely on user–item interaction information and do not explicitly model user–user social relations.

Social recommendation methods such as EATNN, EAGCN, and EGCL generally achieve better results by incorporating social information into user representation learning. Among them, EGCL performs competitively due to its graph adaptive propagation and contrastive learning mechanisms. However, RIBGCL further improves upon EGCL by introducing confidence-gated social denoising and counterfactual item debiasing. This indicates that merely incorporating social relations is not sufficient; it is also important to distinguish reliable social relations from noisy ones and suppress biased item-side signals.

4.2.3 Performance under Different Embedding Dimensions

To examine the effect of representation capacity, Figures 10a–b compare the reported Precision@10 and NDCG@10 for embedding dimensions 32 and 64. These plots report dimensional sensitivity, not user-activity cohorts, and consequently cannot be used as evidence of cold-start robustness.

For both displayed dimensions, the proposed method has the largest bars among the four models shown. Changing the embedding dimension from 32 to 64 produces only a moderate change in the plotted metrics. Since no per-activity-group numerical results are available in the supplied experiment package, no claims about performance for low-, medium-, or high-activity user cohorts are made here.

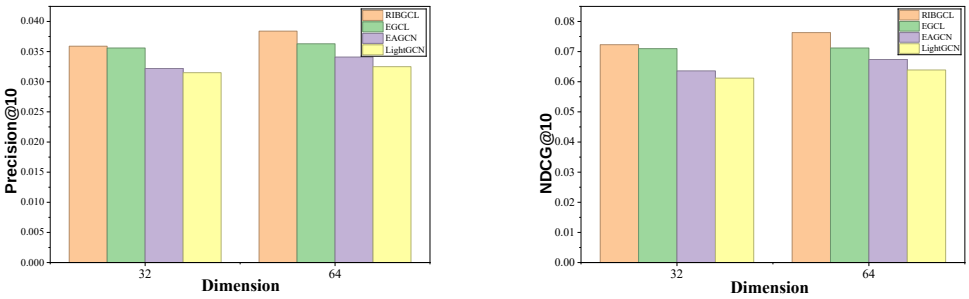


Fig. 10 Reported ranking metrics for different embedding dimensions on Ciao: (a) Precision@10; (b) NDCG@10. The vertical axes display metric fractions, not percentages.

4.3 Training Efficiency Analysis

Table 5 reports the training efficiency comparison of different methods on Epinions and Ciao. Here, "E" denotes the average time cost per epoch, and "T" denotes the total training time.

Table 5 Training efficiency comparison on Epinions and Ciao.

Method	Epinions E(s)	Epinions T(s)	Ciao E(s)	Ciao T(s)
BPRMF	1.30	391.27	0.95	283.61
LightGCN	21.17	10584.83	12.58	6289.42
NGCF	12.40	1239.83	7.37	736.72
EATNN	10.14	10143.86	16.51	16514.49
EAGCN	1.00	3008.37	1.00	3007.21
EGCL	1.54	3064.35	2.35	4706.65
RIBGCL	1.77	3540.23	1.97	3949.06

As shown in Table 5, BPRMF has the lowest training time due to its simple matrix factorization structure. However, its recommendation accuracy is significantly lower than that of graph-based and social-aware methods. LightGCN and EATNN require longer training time on some datasets, indicating that their graph propagation or social modeling processes involve relatively high computational overhead.

Compared with EGCL, RIBGCL introduces additional social confidence gating and counterfactual item debiasing modules. On Epinions, the average time per epoch increases from 1.54 seconds to 1.77 seconds, and the total training time increases from 3064.35 seconds to 3540.23 seconds. This additional computational cost is acceptable considering the consistent performance improvement achieved by RIBGCL.

On Ciao, RIBGCL has a lower average time per epoch and total training time than EGCL. This may be related to sparse matrix computation efficiency, convergence behavior, and implementation differences under different data distributions.

Overall, RIBGCL maintains competitive training efficiency while achieving superior recommendation performance. The proposed modules improve recommendation quality without introducing excessive computational overhead.

4.4 Ablation Study

In this section, several ablation and analytical experiments were conducted to evaluate the effectiveness of the proposed modules. First, component-level ablation was performed to examine the contribution of the social confidence gating module and the counterfactual item debiasing module. Then, the effect of social relation filtering was analyzed to compare hard filtering with dynamic social confidence modeling. After that, the influence of counterfactual debiasing strength was studied by varying the value of `cf_scale`. Finally, social relation quality was analyzed from the perspective of user interaction overlap to further explain the necessity of modeling social confidence.

4.4.1 Component Ablation

To evaluate the contribution of each proposed component, ablation experiments were conducted on the Ciao dataset. The following variants were compared:

- **w/o Social Gate**: The social confidence gating module was removed, and the original social graph was directly used for social propagation.
- **w/o CF Debiasing**: The counterfactual item debiasing module was removed.
- **w/o CF Debiasing & w/o Social Gate**: Both the counterfactual item debiasing module and the social confidence gating module were removed.
- **RIBGCL**: The complete model with both proposed modules.

The ablation results are shown in Table 6.

Table 6 Ablation study on the Ciao dataset.

Metric	w/o SG	w/o CF	w/o Both	RIBGCL
Precision@10	0.0372	0.0378	0.0363	**0.0388**
Recall@10	0.0869	0.0899	0.0839	**0.0931**
MAP@10	0.0425	0.0452	0.0412	**0.0462**
NDCG@10	0.0735	0.0751	0.0711	**0.0781**

In Table 6, SG denotes the social confidence gating module, and CF denotes the counterfactual item debiasing module. As shown in Table 6, the complete RIBGCL model achieves the best performance on all four evaluation metrics. Specifically, RIBGCL achieves 0.0388 on Precision@10, 0.0931 on Recall@10, 0.0462 on MAP@10, and 0.0781 on NDCG@10. These results verify the effectiveness of the proposed modules.

When the social confidence gating module was removed, the performance decreased on all metrics. Compared with RIBGCL, the variant w/o SG decreases from 0.0388 to 0.0372 on Precision@10, from 0.0931 to 0.0869 on Recall@10, from 0.0462 to 0.0425 on MAP@10, and from 0.0781 to 0.0735 on NDCG@10. This indicates that directly using the original social graph may introduce noisy social information into user representation learning.

When the counterfactual item debiasing module was removed, performance degradation was also observed. The variant w/o CF obtains 0.0378 on Precision@10, 0.0899

on Recall@10, 0.0452 on MAP@10, and 0.0751 on NDCG@10. These values are lower than those of the complete model, indicating that item-side bias can affect recommendation performance and that counterfactual debiasing is beneficial for learning more reliable item representations.

When both modules were removed, the performance became the worst among all variants. The variant w/o Both only achieves 0.0363 on Precision@10, 0.0839 on Recall@10, 0.0412 on MAP@10, and 0.0711 on NDCG@10. This further confirms that the two proposed modules contribute from different perspectives and are complementary to each other.

4.4.2 Effect of Social Relation Filtering

To further evaluate the influence of social relation filtering, an additional experiment was conducted on the Ciao dataset. Three variants were compared: Full Social Graph, Low-confidence Removed, and RIBGCL. In the Full Social Graph variant, all original social relations were directly used for social propagation. In the Low-confidence Removed variant, low-confidence social edges were removed before social propagation. In RIBGCL, the proposed social confidence gating module was used to dynamically assign different weights to social neighbors instead of simply removing edges.

The results are shown in Table 7.

Table 7 Effect of social relation filtering on the Ciao dataset.

Variant	Precision@10	Recall@10	MAP@10	NDCG@10
Full Social Graph	0.0372	0.0869	0.0425	0.0735
Low-confidence Removed	0.0381	0.0912	0.0448	0.0764
RIBGCL	**0.0388**	**0.0931**	**0.0462**	**0.0781**

As shown in Table 7, directly using the full social graph obtains 0.0372 on Precision@10, 0.0869 on Recall@10, 0.0425 on MAP@10, and 0.0735 on NDCG@10. After low-confidence social relations were removed, the performance improves to 0.0381 on Precision@10, 0.0912 on Recall@10, 0.0448 on MAP@10, and 0.0764 on NDCG@10. This result indicates that low-confidence social edges can introduce noisy information into social representation learning.

However, RIBGCL achieves the best performance on all metrics, with 0.0388 on Precision@10, 0.0931 on Recall@10, 0.0462 on MAP@10, and 0.0781 on NDCG@10. Compared with simply removing low-confidence edges, RIBGCL further improves the performance. This shows that dynamically weighting social relations is more effective than hard filtering. The proposed social confidence gating module can preserve potentially useful social information while reducing the influence of unreliable social relations.

4.4.3 Effect of Counterfactual Debiasing Strength

To further analyze the influence of counterfactual debiasing, experiments were conducted with different values of `cf_scale`. The parameter `cf_scale` controls the

strength of the counterfactual debiasing signal. A smaller value may provide insufficient debiasing, while a larger value may over-suppress useful item preference information. Therefore, four values, namely 0.01, 0.05, 0.10, and 0.20, were compared on the Ciao dataset.

The results are shown in Table 8.

Table 8 Effect of counterfactual debiasing strength on the Ciao dataset.

`cf_scale`	**Precision@10**	**Recall@10**	**MAP@10**	**NDCG@10**
0.01	0.03862	0.09171	0.0461	0.0774
0.05	0.03876	0.0914	0.0455	0.0771
0.10	**0.03882**	**0.0931**	**0.0462**	**0.0781**
0.20	0.0384	0.0907	0.0455	0.0768

As shown in Table 8, the recommendation performance is affected by the value of `cf_scale`. When `cf_scale` is too small, the counterfactual debiasing effect may be limited, and biased item signals may still remain in the learned representations. When `cf_scale` is too large, useful collaborative filtering information may be excessively suppressed, which can also harm recommendation accuracy.

Therefore, an appropriate value of `cf_scale` is important for balancing preference preservation and bias suppression. The results demonstrate that counterfactual debiasing strength should be carefully controlled rather than simply increased.

4.4.4 Social Relation Quality Analysis

To further explain why social confidence modeling is necessary, the quality of social relations in the original social graphs was analyzed on three datasets, including Ciao, Epinion, and FilmTrust. For each social edge, the interaction overlap between the two connected users was measured by Jaccard similarity. A larger value means that two socially connected users have more common interacted items, while a smaller value indicates weaker interest consistency.

For each dataset, all valid social edges were grouped according to their Jaccard similarity scores. The low group contains social edges with the lowest similarity scores, the middle group contains edges with medium similarity scores, and the high group contains edges with the highest similarity scores. The results are shown in Table 9.

As shown in Table 9, social relations in the original graphs show clearly different levels of interest consistency. On Ciao, the low group has an average Jaccard similarity of 0.000000, while the high group reaches 0.062244. A similar trend can also be observed on Epinion, where the low group has an average Jaccard similarity of 0.000000 and the high group reaches 0.021116. These results indicate that many social edges connect users with very limited or no interaction overlap.

On FilmTrust, the overall Jaccard similarities are higher than those on Ciao and Epinion. However, a clear gap still exists between different groups. The low group has an average Jaccard similarity of 0.006767, while the high group reaches 0.477576. This indicates that even in a relatively denser or more consistent social graph, social relations still have different reliability levels.

Table 9 Social relation quality analysis on different datasets.

Dataset	Group	Avg. Jaccard	Num. Edges
Ciao	Low 20%	0.000000	56830
Ciao	Middle 60%	0.015947	32482
Ciao	High 20%	0.062244	22469
Epinion	Low 20%	0.000000	10812
Epinion	Middle 60%	0.005520	3737
Epinion	High 20%	0.021116	3653
FilmTrust	Low 20%	0.006767	327
FilmTrust	Middle 60%	0.111751	978
FilmTrust	High 20%	0.477576	327

Overall, these results suggest that social relations are not equally reliable across datasets. Some social edges connect users with similar interests, while many others connect users with weak interaction consistency. If all social relations are directly used for information propagation, unreliable neighbor preferences may be introduced into user representations.

Therefore, this analysis provides data-level evidence for the proposed social confidence gating module. Instead of treating all social neighbors equally, the module was designed to adaptively estimate the reliability of social relations and reduce the influence of noisy social connections during social representation learning.

5 Conclusion and Future Work

This study presented RIBGCL, a graph contrastive social recommendation framework with two complementary additions to an established adaptive-propagation backbone: residual confidence weighting of social edges and bounded neighborhood-guided refinement of item embeddings. Specifically, the social confidence gating module dynamically estimates social edge reliability using interest-domain user similarity, constructs a confidence-weighted social graph through a residual confidence design, and effectively reduces the interference of noisy social relations on user representation learning. The counterfactual item debiasing module generates item-side counterfactual vectors from the user neighborhood and attenuates selected directions with a weak residual strategy, thereby limiting changes to the original collaborative representation. The analysis establishes a monotone, residual-floor gate and a norm-bounded item perturbation, clarifying the types of control supplied by the two components. Experiments on three public datasets report improvements in Top-10 ranking accuracy over the included baselines, and ablations support the utility of each added module. However, direct bias metrics, controlled social-noise injection tests, uncertainty estimates, and up-to-date head-to-head baseline runs are needed for stronger claims about debiasing and robustness.

In future work, two directions are planned for further improvement of RIBGCL. First, beyond graph-structured data, it is of interest to explore how unstructured data, such as visual and textual information, can be integrated into the framework to construct more robust multimodal recommender systems (Xu et al. 2026; Malitesta et al. 2026; Zhou and Shen 2023; Chen et al. 2026; Ding et al. 2026; Ahn et al. 2026; Choi

et al. 2025; Yuan et al. 2023).Second, the current framework captures user preferences based on a single type of user–item interaction, whereas real-world user behaviors are inherently diverse, such as likes, collections, and shares. Furthermore, with the development of large language models and agent technologies, LLM-based recommendation, zero-shot ranking, and interactive information retrieval also provide new research directions for social recommendation(Bao et al. 2023; Lin et al. 2025; Hou et al. 2024; Deldjoo et al. 2025; Pomo et al. 2025; Jia et al. 2025). Therefore, incorporating multiple behavioral signals into RIBGCL to achieve a more comprehensive understanding of user intent is an interesting research direction.

Statements and Declarations

Funding.

This work was supported in part by the Shanghai Science and Technology Innovation Action Plan (Grant No. 22ZR1425300).

Competing interests.

The authors declare that they have no competing interests relevant to the content of this article.

Ethics approval.

Ethics approval was not required because the study analyzed existing public benchmark datasets without participant recruitment or intervention.

Consent to participate and consent for publication.

Not applicable.

Data availability.

The Epinions, Ciao, and FilmTrust benchmark data used in this study are publicly available from the sources described in the Experimental Setup. The public dataset entry points are given in Sect. 4.

Code Availability.

The source code is available at https://github.com/Abbott-1/RIBGCL.git.

Author contributions.

Yinqi Zhang: conceptualization, experiments, analysis, writing—original draft. Peilin Yang: supervision, methodology review, writing—review and editing. Xiaogang Wang: manuscript review. Ruixu Wu: revision and results checking.

References

Ahn S, Shin S, Seo YD (2026) Enriching semantic profiles into knowledge graph for recommender systems using large language models. In: Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp 25–36

Anand V, Maurya AK (2025) A survey on recommender systems using graph neural network. ACM Transactions on Information Systems 43(1):1–49

Bao K, Zhang J, Zhang Y, et al (2023) Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In: Proceedings of the 17th ACM Conference on Recommender Systems. ACM, New York, NY, USA, pp 1007–1014

Cai X, Huang C, Xia L, et al (2023) Lightgcl: Simple yet effective graph contrastive learning for recommendation. In: Proceedings of the International Conference on Learning Representations, URL https://arxiv.org/abs/2302.08191, arXiv:2302.08191

Chen C, Zhang M, Liu Y, et al (2019a) Social attentional memory network: Modeling aspect- and friend-level differences in recommendation. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. ACM, New York, NY, USA, pp 177–185

Chen C, Zhang M, Liu Y, et al (2019b) Social recommendation with dual graph attention networks. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, NY, USA, pp 985–988

Chen C, Zhang M, Wang C, et al (2019c) An efficient adaptive transfer neural network for social-aware recommendation. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 225–234, https://doi.org/10.1145/3331184.3331192

Chen C, Ma W, Zhang M, et al (2021a) Graph heterogeneous multi-relational recommendation. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence. AAAI Press, Palo Alto, CA, USA, pp 3958–3966

Chen J, Dong H, Qiu Y, et al (2021b) Autodebias: Learning to debias for recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, NY, USA, pp 21–30

Chen J, Zhang F, Li H, et al (2026) M4rec: Multi-modal knowledge graph modeling of multi-dimensional user preferences for next-poi recommendation. IEEE Transactions on Knowledge and Data Engineering 38(6):3751–3764. https://doi.org/10.1109/TKDE.2026.3677245

Chen M, Wei Z, Huang Z, et al (2020) Simple and deep graph convolutional networks. In: Proceedings of the 37th International Conference on Machine Learning. PMLR, Virtual Event, pp 1725–1735

Chen Y, Wu L, Zaki MJ (2021c) Knowledge-aware coupled graph neural network for social recommendation. In: Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence. AAAI Press, Palo Alto, CA, USA, pp 4115–4122

Choi M, Lee S, Park S, et al (2025) Linear item-item models with neural knowledge for session-based recommendation. In: Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 1666–1675

Deldjoo Y, Sanner S, Palumbo E, et al (2025) Gennext: The next generation of ir and recommender systems with language agents, generative models, and conversational ai. In: Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 4195–4198

Ding H, Zhu W, Shu Y (2026) Hgmrec: A hypergraph-based medical knowledge-enhanced literature recommendation model. IEEE Transactions on Big Data 12(4):1344–1357. https://doi.org/10.1109/TBDATA.2026.3679457

Fan W, Ma Y, Li Q, et al (2019) Graph neural networks for social recommendation. In: Proceedings of the World Wide Web Conference. ACM, New York, NY, USA, pp 417–426

Feng Y, Yang Z, Li K, et al (2026) Attacks and detections in recommender systems: A comprehensive analysis for models, progresses, and trends. IEEE Transactions on Knowledge and Data Engineering 38(2):889–910. https://doi.org/10.1109/TKDE.2025.3639434

Gao C, Zheng Y, Li N, et al (2023) A survey of graph neural networks for recommender systems: Challenges, methods, and directions. ACM Transactions on Recommender Systems 1(1):1–51

Gu Y, Ding S, Li Y, et al (2026) Graph neural networks with dynamic similarity fusion for social recommendation. International Journal of Machine Learning and Cybernetics 17:10. https://doi.org/10.1007/s13042-025-02889-2, URL https://doi.org/10.1007/s13042-025-02889-2

Guo G, Zhang J, Yorke-Smith N (2015) Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. AAAI Press, Palo Alto, CA, USA, pp 123–129

Guo Q, Yang X, Liu K, et al (2026) Controllable exploration-exploitation in user-item interactions toward long-term recommendation gains: A pane-aware graph transformer architecture. IEEE Transactions on Knowledge and Data Engineering 38(7):4185–4199. https://doi.org/10.1109/TKDE.2026.3689197

He X, Liao L, Zhang H, et al (2017) Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web. ACM, New York, NY, USA, pp 173–182

He X, Deng K, Wang X, et al (2020) Lightgcn: Simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, NY, USA, pp 639–648

Hou Y, Zhang J, Lin Z, et al (2024) Large language models are zero-shot rankers for recommender systems. In: Proceedings of the European Conference on Information Retrieval. Springer, Cham, Switzerland, pp 364–381

Hu Y, Koren Y, Volinsky C (2008) Collaborative filtering for implicit feedback datasets. In: Proceedings of the 8th IEEE International Conference on Data Mining. IEEE, Piscataway, NJ, USA, pp 263–272

Hu Z, Nakagawa S, Zhuang Y (2025) Hierarchical denoising for robust social recommendation. IEEE Transactions on Knowledge and Data Engineering 37(2):739–753

Huang J, Chen J, Lin J, et al (2025a) A comprehensive survey on retrieval methods in recommender systems. ACM Transactions on Information Systems 44(1):1–43

Huang T, Dong Y, Ding M, et al (2021) Mixgcf: An improved training method for graph neural network-based recommender systems. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, New York, NY, USA, pp 665–674

Huang Y, Liang K, Huang Y, et al (2025b) Social recommendation via graph-level counterfactual augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 334–342

Jia P, Cai Q, Zhao X, et al (2025) Agentir: 2nd workshop on agent-based information retrieval. In: Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 4180–4183

Jiang Y, Huang C, Huang L (2023) Adaptive graph contrastive learning for recommendation. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, New York, NY, USA, pp 4252–4261

Khan B, Wu J, Yang J, et al (2025) Heterogeneous hypergraph neural network for social recommendation using attention network. ACM Transactions on Recommender Systems 3(3):1–22

Li A, Yang B (2025) Dual graph denoising model for social recommendation. In: Proceedings of the ACM Web Conference. ACM, New York, NY, USA, pp 347–356

Li J, Guo J, Wang L (2026a) Noise-tolerant graph neural network-based social recommendation algorithm. International Journal of Machine Learning and Cybernetics 17:351. https://doi.org/10.1007/s13042-026-03179-1, URL https://doi.org/10.1007/s13042-026-03179-1

Li Y, Ning X, Zhu Z, et al (2026b) Knowledge graph-based debiasing for trustworthy recommendation systems. IEEE Transactions on Knowledge and Data Engineering 38(8):5232–5244. https://doi.org/10.1109/TKDE.2026.3702349

Li Z, Xia L, Huang C (2024) Recdiff: Diffusion model for social recommendation. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, pp 1346–1355, https://doi.org/10.1145/3627673.3679630

Lin X, Bao K, Zhang J, et al (2025) Navigating large language models for recommendation: From architecture to learning paradigms and deployment. In: Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 4134–4137

Liu C, Zhang J, Wang S, et al (2025) Score-based generative diffusion models for social recommendations. IEEE Transactions on Knowledge and Data Engineering 37(11):6666–6679. https://doi.org/10.1109/TKDE.2025.3600103

Liu F, Cheng Z, Zhu L, et al (2021) Interest-aware message-passing gcn for recommendation. In: Proceedings of the Web Conference 2021. ACM, New York, NY, USA, pp 1296–1305

Ma H, Zhou D, Liu C, et al (2011) Recommender systems with social regularization. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining. ACM, New York, NY, USA, pp 287–296

Malitesta D, Rossi E, Pomo C, et al (2026) Training-free graph-based imputation of missing modalities in multimodal recommendation. IEEE Transactions on Knowledge and Data Engineering 38(5):3250–3263. https://doi.org/10.1109/TKDE.2026.3667005

Ning L, Fan W, Li Q (2026) Exploring backdoor attack and defense for llm-empowered recommendations. IEEE Transactions on Knowledge and Data Engineering 38(8):5077–5094. https://doi.org/10.1109/TKDE.2026.3690651

Peng T, et al (2025) Tagrec: Temporal-aware graph contrastive learning with theoretical augmentation for sequential recommendation. IEEE Transactions on Knowledge and Data Engineering 37(5):3015–3029. https://doi.org/10.1109/TKDE.2025.3538706

Pires JM, Silva EFd, Durão FA (2026) Exploiting distribution-based confidence integration in graph neural network recommenders. Applied Intelligence 56:142. https://doi.org/10.1007/s10489-026-07187-0

Pomo C, Jannach D, Kim Y, et al (2025) First international workshop on data quality-aware multimodal recommendation. In: Proceedings of the Nineteenth ACM Conference on Recommender Systems. ACM, New York, NY, USA, pp 1378–1382

Purificato A, Cassarà G, Siciliano F, et al (2025) Sheaf4rec: Sheaf neural networks for graph-based recommender systems. ACM Transactions on Recommender Systems 4(2):1–26

Ren X, Wei W, Xia L, et al (2026) A comprehensive survey on self-supervised learning for recommendation. ACM Computing Surveys 58(1):1–38. https://doi.org/10.1145/3746280

Rendle S, Freudenthaler C, Gantner Z, et al (2009) Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. AUAI Press, Arlington, VA, USA, pp 452–461

Rodríguez-Baena D, Gómez-Vela F, Lopez-Fernandez A, et al (2025) Binrec: Addressing data sparsity and cold-start challenges in recommender systems with biclustering. Applied Intelligence 55:830. https://doi.org/10.1007/s10489-025-06725-6

Sang L, Wang Y, Zhang Y, et al (2025a) Intent-guided heterogeneous graph contrastive learning for recommendation. IEEE Transactions on Knowledge and Data Engineering 37(4):1915–1929. https://doi.org/10.1109/TKDE.2025.3536096

Sang S, Xue F, Li S, et al (2025b) Acd: Adversarial counterfactual distillation for rating prediction in recommendation. IEEE Transactions on Knowledge and Data Engineering 38(2):814–826

Tang S, Lin S, Ma J, et al (2025) Coder: Counterfactual demand reasoning for sequential recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 12649–12657

Wang F, Qi L, Liu W, et al (2025) Inter- and intra-similarity preserved counterfactual incentive effect estimation for recommendation systems. ACM Transactions on Information Systems 43(6):1–24

Wang M, Li W, Shi J, et al (2023) Dor: A novel dual-observation-based approach for recommendation systems. Applied Intelligence 53:29109–29127. https://doi.org/10.1007/s10489-023-05075-5

Wang X, He X, Wang M, et al (2019) Neural graph collaborative filtering. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, NY, USA, pp 165–174

Wang X, Jin H, Zhang A, et al (2020) Disentangled graph collaborative filtering. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and

Development in Information Retrieval. ACM, New York, NY, USA, pp 1001–1010

Wei T, Wang Le, Fu X, et al (2025) User preference representation and dual contrastive learning for knowledge-aware recommendation. International Journal of Machine Learning and Cybernetics 16:9441–9456. https://doi.org/10.1007/s13042-025-02763-1, URL https://doi.org/10.1007/s13042-025-02763-1

Wu B, Zhong L, Yao L, et al (2022a) Eagcn: An efficient adaptive graph convolutional network for item recommendation in social internet of things. IEEE Internet of Things Journal 9(17):16386–16401. https://doi.org/10.1109/JIOT.2022.3151400

Wu B, Zhang B, Tian Y, et al (2026) Egcl: An effective and efficient graph contrastive learning framework for social recommendation. ACM Transactions on Information Systems 44(3):1–30. https://doi.org/10.1145/3786595

Wu J, Wang X, Feng F, et al (2021) Self-supervised graph learning for recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, NY, USA, pp 726–735

Wu J, Wang X, Gao X, et al (2022b) Towards representation alignment and uniformity in collaborative filtering. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, New York, NY, USA, pp 2124–2134

Xiong F, Zhang T, Pan S, et al (2025) Robust graph based social recommendation through contrastive multi-view learning. Proceedings of the AAAI Conference on Artificial Intelligence 39(12):12890–12898. https://doi.org/10.1609/aaai.v39i12.33406

Xu J, Chen Z, Yang S, et al (2025) Nlgcl: Naturally existing neighbor layers graph contrastive learning for recommendation. In: Proceedings of the Nineteenth ACM Conference on Recommender Systems. ACM, New York, NY, USA, pp 319–329, https://doi.org/10.1145/3705328.3748059

Xu J, Chen Z, Yang S, et al (2026) A survey on multimodal recommender systems: Recent advances and future directions. IEEE Transactions on Multimedia pp 1–14. https://doi.org/10.1109/TMM.2026.3668620

Yang B, Lei Y, Liu D, et al (2017) Social collaborative filtering by trust. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(8):1633–1647

Yang K, Li G, Wang L, et al (2025a) A framework for solving bias in graph-based recommender systems. Applied Intelligence 55(7). https://doi.org/10.1007/s10489-025-06388-3

Yang K, Li G, Wang L, et al (2025b) A framework for solving bias in graph-based recommender systems with a causal perspective. Applied Intelligence 55:475. https://doi.org/10.1007/s10489-025-06388-3

Yang Y, Huang C, Xia L, et al (2023) Debiased contrastive learning for sequential recommendation. In: Proceedings of the ACM Web Conference. ACM, New York, NY, USA, pp 1063–1073

Ying R, He R, Chen K, et al (2018) Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York, NY, USA, pp 974–983

Yu J, Gao M, Li J, et al (2021a) Adaptive implicit friends identification over heterogeneous network for social recommendation. In: Proceedings of the 30th ACM International Conference on Information and Knowledge Management. ACM, New York, NY, USA, pp 3578–3582

Yu J, Yin H, Li J, et al (2021b) Self-supervised multi-channel hypergraph convolutional network for social recommendation. In: Proceedings of The Web Conference 2021. ACM, New York, NY, USA, pp 413–424

Yu J, Yin H, Xia X, et al (2022) Are graph augmentations necessary? simple graph contrastive learning for recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, NY, USA, pp 1294–1303

Yu J, Xia X, Chen T, et al (2023) Xsimgcl: Towards extremely simple graph contrastive learning for recommendation. IEEE Transactions on Knowledge and Data Engineering 36(2):913–926

Yu W, Qin Z (2020) Graph convolutional network for recommendation with low-pass collaborative filters. In: Proceedings of the 37th International Conference on Machine Learning. PMLR, Virtual Event, pp 10936–10945

Yu Y, Sugiyama K, Jatowt A (2025) Domain counterfactual data augmentation for explainable recommendation. ACM Transactions on Information Systems 43(3):1–30

Yuan Z, Yuan F, Song Y, et al (2023) Where to go next for recommender systems? id-vs. modality-based recommender models revisited. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 2639–2649

Zeng H, Wang Le, Li S, et al (2026) Multi-side contrastive learning with intent-enhanced for knowledge recommendation. International Journal of Machine Learning and Cybernetics 17:194. https://doi.org/10.1007/s13042-026-03021-8, URL https://doi.org/10.1007/s13042-026-03021-8

Zhang K, Cao Q, Sun F, et al (2025a) Robust recommender system: A survey and future directions. ACM Computing Surveys 58(1):1–38

Zhang L, Ke M, Wu L, et al (2025b) Multi-objective graph contrastive learning for recommendation. IEEE Transactions on Big Data 11(5):2696–2709. https://doi.org/10.1109/TBDATA.2025.3552341

Zhang Q, Yang P, Yu J, et al (2025c) A survey on point-of-interest recommendation: Models, architectures, and security. IEEE Transactions on Knowledge and Data Engineering 37(6):3153–3172. https://doi.org/10.1109/TKDE.2025.3551292

Zhou X, Shen Z (2023) A tale of two graphs: Freezing and denoising graph structures for multimodal recommendation. In: Proceedings of the 31st ACM International Conference on Multimedia. ACM, New York, NY, USA, pp 935–943