SoREX: Towards Self-Explainable Social Recommendation with Relevant Ego-Path Extraction

HANZE GUO*, Gaoling School of Artificial Intelligence, Renmin University of China, China
YIJUN MA*, Gaoling School of Artificial Intelligence, Renmin University of China, China
XIAO ZHOU†‡§, Gaoling School of Artificial Intelligence, Renmin University of China, China

Social recommendation has been proven effective in addressing data sparsity in user-item interaction modeling by leveraging social networks. The recent integration of Graph Neural Networks (GNNs) has further enhanced prediction accuracy in contemporary social recommendation algorithms. However, many GNN-based approaches in social recommendation lack the ability to furnish meaningful explanations for their predictions. In this study, we confront this challenge by introducing SoREX, a self-explanatory GNN-based social recommendation framework. SoREX adopts a two-tower framework enhanced by friend recommendation, independently modeling social relations and user–item interactions, while jointly optimizing an auxiliary task to reinforce social signals. To offer explanations, we propose a novel ego-path extraction approach. This method involves transforming the ego-net of a target user into a collection of multi-hop ego-paths, from which we extract factor-specific and candidate-aware ego-path subsets as explanations. This process facilitates the summarization of detailed comparative explanations among different candidate items through intricate substructure analysis. Furthermore, we conduct explanation re-aggregation to explicitly correlate explanations with downstream predictions, imbuing our framework with inherent self-explainability. Comprehensive experiments conducted on four widely adopted benchmark datasets validate the effectiveness of SoREX in predictive accuracy. Additionally, qualitative and quantitative analyses confirm the efficacy of the extracted explanations in SoREX. Our code and data are available at https://github.com/antman9914/SoREX.

CCS Concepts: • **Information systems** → **Social recommendation**; **Recommender systems**; • **Computing methodologies** → **Neural networks**.

Additional Key Words and Phrases: Social Recommendation, Graph Neural Networks, Explainable Recommendation.

ACM Reference Format:
Hanze Guo, Yijun Ma, and Xiao Zhou. 2018. SoREX: Towards Self-Explainable Social Recommendation with Relevant Ego-Path Extraction. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 27 pages. https://doi.org/XXXXXXX.XXXXXXX

*Both authors contributed equally to this research.
†Corresponding author
‡Also with Beijing Key Laboratory of Research on Large Models and Intelligent Governance.
§Also with Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE.

Authors' Contact Information: Hanze Guo, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China, ghz@ruc.edu.cn; Yijun Ma, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China, mayj_hedgehog@ruc.edu.cn; Xiao Zhou, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China, xiaozhou@ruc.edu.cn.

1 Introduction

Recommender systems have become a common choice for alleviating information overload after the prosperity of the Internet [12, 23]. However, they always suffer from the sparse interactions between users and items. According to social correlation theories [2], it is believed that users' preferences can be influenced by their social relationships. With the proliferation of online social platforms, social recommendation has been designed to improve user-item interaction modeling, addressing data sparsity and cold start problem with the aid of user-user social regularizations [16, 53].

Recently, Graph Neural Networks (GNNs) [11, 71] have been widely adopted in recommender systems owing to their robust ability to model structured data. Notably, contemporary social recommendation algorithms [14, 29, 40, 48, 61] have embraced GNNs to harness high-order social context and collaborative information simultaneously. Despite the significant accuracy enhancements brought about by GNN-based methods, they often fall short in providing meaningful explanations for their predictions.

However, the explainability of predictions made by social recommender systems is of paramount importance for both users and service providers. For users, explainability serves as a foundation for fostering engagement and trust in the system [66], enabling them to make more informed decisions [34]. Online A/B testing on e-commerce platforms has demonstrated that providing explanations can improve click-through rates in real-world commercial settings [66]. For service providers, explainability is equally critical. High-profile incidents, such as YouTube's loss of advertising revenue due to the opaque nature of its recommendation algorithms [13], and Amazon's discontinuation of a biased recruitment system [5], underscore the potential risks and consequences of lacking transparency. Therefore, enhancing the explainability of recommendation models not only improves system transparency but also aids in uncovering systematic patterns, thereby deepening our understanding of underlying network characteristics [65].

Current research on explainable recommendation systems often draws on user reviews [1, 39, 42] and knowledge graphs (KGs) [44, 50, 62]. Review-based methods focus on generating coherent and persuasive explanations for users, but they heavily depend on the availability and quality of textual reviews [1]. In many scenarios, such reviews may be sparse, biased, or even absent, especially for newly added items or users, which greatly limits the generalizability of these methods. On the other hand, KG-based methods aim to improve transparency by leveraging semantic paths within well-constructed knowledge graphs [62]. However, such graphs are often unavailable or incomplete in practice, particularly in social networks or user–item interaction graphs, which restricts the applicability of KG-based approaches.

To overcome the limitations of existing explanation methods, we emphasize substructure mining [45, 57]—a widely used graph-based interpretability technique that extracts the most contextually relevant and interpretable subgraphs. This approach helps uncover complex, factor-specific patterns essential for understanding the reasoning behind recommendation rankings. In this work, we leverage the structural properties of user–item interaction graphs and social networks to enhance the transparency of social recommender systems, enabling more general and broadly applicable explanations. However, most substructure mining methods are post hoc [27, 36, 57], generating explanations only after model training. Such approaches are prone to distribution shifts [64, 69], which can compromise explanation fidelity. As a result, recent efforts have focused on developing self-explainable GNNs [4, 7, 30, 45] that generate explanatory subgraphs as part of the prediction process. Nevertheless, existing methods primarily target graph-level classification and often fail to capture the complex, node-pair-specific substructures required for recommendation tasks. Moreover, explanations in recommendation typically answer the question, "Why should this user choose this item?"—yet in ranking-based settings, more persuasive explanations may arise from comparative reasoning, i.e., "Why should this

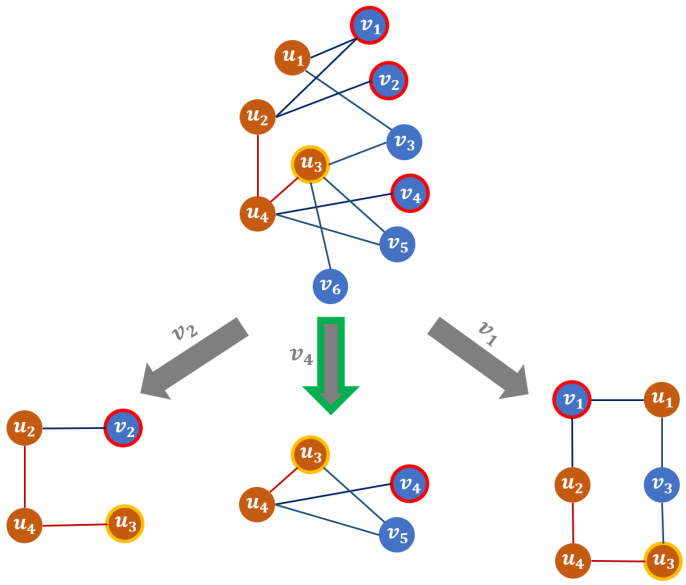


Fig. 1. A toy example illustrating a substructure-based comparative explanation for recommending item v_4 to user u_3. We enumerate all meta-paths of length no longer than 3 from u_3 to items v_1, v_2, and v_4. While v_1 and v_4 each have two paths and v_2 only one, v_4 is more relevant due to a shorter, socially meaningful path via direct friend u_4. This highlights how structural signals alone can yield accurate and interpretable recommendations.

item be ranked higher than others?" While [51] attempts to address this question, it relies heavily on textual reviews and does not explicitly improve model transparency.

Therefore, two major challenges remain: **(1) Pair-wise complexity**: While graph-based explanation methods excel at providing node-level interpretability, they often struggle to capture the pair-wise interpretability essential for recommendation tasks, where understanding the relationship between specific user-item pairs is crucial; **(2) Comparative explanations**: Existing methods rarely provide effective comparative insights between different items in the recommendation ranking, limiting their ability to answer not just 'why this item?' but also 'why this item over others?', which is vital for persuasive and transparent recommendations. Nevertheless, our proposed method can effectively overcome both of these challenges by extracting and aggregating candidate-aware ego-path subgraphs for each user–item pair, and supporting direct comparison among candidates for transparent, comparative ranking explanations. As illustrated in Fig. 1, we can *solely* leverage the structural user–item and social graph—without reviews, textual KG facts or other hard-to-obtain signals—by jointly evaluating the number and the quality of meta-paths of length ≤ 3; in this toy case, this structural criterion already lifts v_4 to the top of the ranking. Such *comparative*, structure-based explanations can be surfaced to both users and developers, improving recommendation accuracy while clarifying the model's decision process.

To fill the comparative explainability gap in GNN-based social recommendation, we propose a novel framework named *SoREX* (*So*cial *R*ecommendation based on relevant *E*go-path e*X*traction). SoREX is designed to be self-explainable, enabling the exploration of candidate-aware complex substructures and comparative relationships among candidate items from different perspectives. It is built on a two-tower architecture: a social influence-aware social tower and an interaction tower, each equipped with separate GNN encoders to learn from the social network and user-item interaction

graph respectively. The towers independently conduct user/item embedding and candidate item ranking, with final rankings obtained by fusing both tower predictions. Such design can independently model social and interaction factors, laying foundation for factor-specific explanations. Multi-task learning is further employed during training, where a friend-recommendation auxiliary task is attached to the social tower to capture more reliable friend relations.

For clarity, we denote users' multi-hop neighborhoods as their *ego-nets*. Since GNNs can only perceive the ego-net of the target user, we propose to extract a subgraph of the ego-net relevant to each individual candidate item and factor, serving as a candidate-aware and factor-specific explanation. To enable complex substructure investigation and provide detailed comparison, we aim to extract dense explanatory subgraphs instead of sparse ones, as suggested by [30, 31]. We achieve this by transforming the ego-net into a set of *ego-paths*, shared by both towers, representing all multi-hop paths on the joint graph of the social network and user-item interaction graph originating from the target user. Random walk sampling ensures memory efficiency during ego-path generation. Next, we compute similarities between the given candidate and all observed ego-paths in each tower. Transforming these similarities into ego-path sampling probabilities, we obtain a subset of ego-paths relevant to each candidate and factor. Finally, explanation re-aggregation aggregates information from the sampled ego-path subsets into the final user representations used for downstream prediction, emphasizing candidate-aware and factor-specific neighborhood information. Ego-paths, forming motifs through interweaving, enable complex substructure investigation. Additionally, each candidate is assigned factor-specific explanation graphs with different structures and similarity distributions, naturally forming comparative relationships and explanations among candidate items.

In summary, our contributions are highlighted as follows:

- We introduce a two-tower GNN-based social recommendation framework that independently models user preferences from social and user-item interaction perspectives.
- We devise a novel explanation extraction strategy that samples factor-specific and candidate-aware subsets of multi-hop ego-paths for each candidate item. This enables detailed comparative explanations for ranking predictions and facilitates high-level substructure analysis.
- We propose explanation re-aggregation to connect explanatory ego-path subsets to predictions, making our framework self-explainable. To the best of our knowledge, we are the first to address the comparative explainability gap in GNN-based social recommendation.
- Empirical experiments on four benchmark datasets substantiate the superiority of SoREX in accuracy. Additionally, we qualitatively and quantitatively demonstrate the explainability of our method.

2 Related Work

2.1 Social Recommendation

Recommender systems have been widely explored as an effective approach for modeling user–item interactions [11, 22, 70]. However, the sparsity of user-item interaction often limits their performance. To alleviate this issue, social recommendation is proposed to incorporate social networks and enrich user–item interaction modeling. Early studies [10, 16, 28, 49, 53] usually use social network for embedding regularization.

Recent works introduce prospering GNNs into social recommendation due to their strong capability in modeling relational data. GraphRec [8] is the pioneer to design social recommender with attention-based GNN. To model the recursive influence diffusion process, Diffnet [48] and Diffnet++ [47] adopt a multi-layer influence propagation architecture to learn the evolution of user preference. MHCN [61] proposes a multi-channel hypergraph convolutional

network to explicitly leverage high-order user relations. DESIGN [40] leverages multiple GNN encoders to learn from different factors, fusing their knowledge via knowledge distillation. SocialLGN [26] conducts message propagation in both social network and interaction graph in each GNN layer for knowledge fusion. ESRF [60] devises a GCN-based deep adversarial social recommendation framework. Jiang et al. [19] identify the low preference homophily among socially connected users, and propose SHaRe to address the issue via social graph rewiring. Recent works also integrate self-supervised learning [14, 46, 59] and graph denoising [21, 25, 54] into GNN-based social recommendation.

Although these methods significantly improve accuracy, they largely overlook interpretability, leaving users unclear about the reasons behind recommendations. This lack of explanation limits transparency, which is critical for real-world applications. To address this, SoREX is designed as a self-explainable GNN-based social recommender that highlights user preferences and social interactions, improving transparency through substructure-aware and comparative explanations.

2.2 Explainable Recommendation

Most explainable recommendation solutions generate explanations based on item profiles, user reviews and knowledge graphs (KG). Profile-based and review-based methods aim to help users understand their decisions via retrieved or generated textual information. Du et al. [6] proposed a taste cluster based self-explainable collaborative filter method, using item tags for cluster description. As for review-based methods, the retrieval-based methods aim to select review text that matches prediction via attention [1], reinforcement learning [43] and other feasible means. A recent work [51] also raises the conception of comparative explanation, but it is also a review-based method. On the other hand, generation-based methods aim to identify important aspects of interacted items [39, 42] or directly apply text generation techniques [24, 52] to synthesize explanations. KG based methods aim to improve the transparency of recommenders. They mostly employ multi-hop path reasoning on KG , and then apply the optimal paths as explanations. The difference among them mainly lies in the reasoning strategy, such as meta-path template based [62], LSTM based [44] and reinforcement learning based [50, 68] methods.

Existing profile-based, review-based, and knowledge graph (KG)-based explanation methods rely heavily on external data, such as user reviews or structured semantic triples. However, such data is often difficult to obtain, incomplete, or unavailable, especially in social recommendation scenarios involving new users or items. This limits their applicability in practice. While some prior work [18] introduces socially-aware explanations, it still fundamentally depends on reviews. In contrast, SoREX enhances transparency by leveraging only the intrinsic structure of user–item and user–user interactions, without requiring any auxiliary information. Furthermore, most KG-based methods focus on semantic reasoning and are ill-suited for modeling social influence. Our approach fills this gap by introducing a graph-based, self-explainable framework for social recommendation that supports comparative explanations based on mined substructures, thereby improving both interpretability and applicability in real-world settings.

2.3 Explainable GNN

The lack of interpretability of GNNs promote the development of GNN explainers. Early GNN explainers are post-hoc, generating posterior substructure-based explanations for trained GNNs. For example, GNNExplainer [57] and PGExplainer [27] generate edge masks or node feature masks to find the significant subgraphs; RGExplainer [36] searches for relevant subgraphs via reinforcement learning; XGNN [63] trains a graph generator to generate explanation graphs; K-FactExplainer [15] proposes a factorized explainer to reflect one-to-many relationships between labels and explanatory substructures. However, recent works [64, 69] prove that post-hoc explainers suffer from distribution shift between explanations and predictions.

Therefore, efforts have been made to develop self-explainable GNNs. SE-GNN [3] is one of the pioneers, using neighbors with the same label as the input node as explanation. ProtGNN [67] and PxGNN [4] identifies representative graph patterns via prototype learning. ConPI [45] and ILP-GNN [72] model similarity between neighbor sets of given node pair and infer the existence of link between them, which is explained by similar neighbors or neighbor pairs. GIB [58], GSAT [30] and LRI [31] all refer to information bottleneck (IB) theory. GIB proposes to train an explanatory graph generator with IB, while GSAT and LRI inject learnable randomness into GNN and sample dense explanatory graphs without spurious correlations. Some recent works have proposed further extensions to IB based methods. For example, PGIB [35] combines prototype learning with IB theory. There is also a group of causal learning based self-explainable GNNs [7, 38] proposing to disentangle causal subgraphs from biased graph as explanations; Besides, a recent work GraphChef [32] proposes to integrate decision tree into the message passing framework of GNNs to achieve self-explainability.

Although some existing GNN-based explainable methods explore complex motifs for explanation, they are primarily designed for classification tasks and lack the ability to extract node pair aware substructures or provide comparative explanations, both of which are essential for ranking-based recommendation. In contrast, SoREX focuses on intrinsic user–item and user–user relationships, making it especially effective for social recommendation. By integrating self-explainable mechanisms that leverage social context and interaction patterns, SoREX fills a key gap in current explainable GNN research. Unlike traditional approaches, it delivers comparative explanations tailored to ranking tasks and does so without relying on external data, enhancing both transparency and practical applicability in real-world systems.

3 Problem formulation

3.1 Graph Based Social Recommendation

We denote user set and item set as $\mathcal{U} = \{u_1, u_2, ..., u_m\}$ and $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ respectively, where m is the number of users and n is the number of items. $\mathcal{G}_r = (\mathcal{U} \cup \mathcal{V}, \mathcal{E}_r)$ denotes the user-item bipartite, where $(u_i, v_j) \in \mathcal{E}_r$ indicates that an interaction exists between user u_i and item v_j. $\mathcal{G}_s = (\mathcal{U}, \mathcal{E}_s)$ denotes the user-user social graph, where $(u_i, u_j) \in \mathcal{E}_s$ indicates that user u_i and u_j have a social relation. We denote the joint graph of $\mathcal{G}_r$ and $\mathcal{G}_s$ as $\mathcal{G} = (\mathcal{U} \cup \mathcal{V}, \mathcal{E}_r \cup \mathcal{E}_s)$. Let user-item interaction matrix and user-user social relation matrix be $\mathbf{R} \in \mathbb{R}^{m \times n}$ and $\mathbf{S} \in \mathbb{R}^{m \times m}$, where $\mathbf{R}_{ij} = 1$ if $(u_i, v_j) \in \mathcal{E}_r$, and $\mathbf{S}_{ij} = 1$ if $(u_i, u_j) \in \mathcal{E}_s$. The adjacency matrix $\mathbf{A}^R$ of undirected user-item bipartite and adjacency matrix $\mathbf{A}$ of $\mathcal{G}$ can be defined as Eq. 1:

$$\mathbf{A}^R = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0} \end{bmatrix}, \mathbf{A} = \begin{bmatrix} \mathbf{S} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0} \end{bmatrix}. \tag{1}$$

We formulate the graph based social recommendation problem as follows:

Definition 1 (Graph based social recommendation): Given $\mathcal{G}_r$ and $\mathcal{G}_s$, the objective of social recommendation is to predict the missing links in $\mathcal{G}_r$, suggesting the top-K disconnected items that the target users are most likely to interact with.

3.2 Self-Explainable Social Recommendation

Generally, self-explainable graph learning methods generate a subgraph for both explanation and downstream prediction. We further treat the explanation graph as an emphasis over relevant subgraph of ego-net in this work to alleviate information loss.

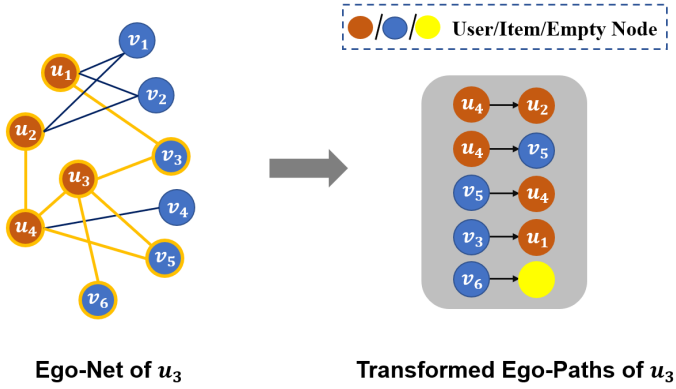


Fig. 2. Example of ego-net and ego-path. The 2-hop ego-net of user u_3 is highlighted on the left, while the ego-net is transformed into a set of 2-hop ego-paths on the right. Given the length of the ego-path passing by v_6 is less than 2, to meet the required length, it is padded with an empty node.

We first define ego-net as follows:

Definition 2 (Ego-net): the k-hop ego-net of user u_i is defined as the k-hop neighborhood of u_i in $\mathcal{G}$, denoted as $\mathcal{G}_{ego}(i,k)$.

Given ego-net $\mathcal{G}_{ego}(i,k)$, we can represent it with the set of all possible paths originating from u_i no longer than k. By removing redundant information, we derive the definition of ego-path:

Definition 3 (Ego-path): Given u_i origined k-hop path $\hat{w}_t(i,k) = \{u_i \rightarrow q_1 \rightarrow ... \rightarrow q_k\}$, by removing the source node u_i and repetitive nodes, we obtain an ego-path $w_t(i,k) = \{q_1 \rightarrow ... \rightarrow q_k\}$ no longer than $k-1$.

The complete set of u_i-origined ego-path can be denoted as $\mathcal{W}_{ego}(i,k) = \{w_t(i,k)\}_{t=1}^{T}$, where T is the amount of ego-paths. The ego-paths shorter than $k-1$ are padded to the required length with an empty node with zero vector embedding. Figure 2 presents a toy example of the relationship between ego-net and ego-path. The yellow empty node indicates padding when there are not enough neighbors, and does not carry any information. We finally define self-explainable social recommendation as follows:

Definition 4 (Self-explainable social recommendation): Given ego-path set $\mathcal{W}_{ego}(i,k)$ of user u_i, for each disconnected item v_j, we extract candidate-aware ego-path subset $\tilde{\mathcal{W}}_{ego}(i,k,j)$, which is expected to be relevant with v_j and able to explain the ranking of v_j. We then predict missing interaction links for u_i based on $\{\tilde{\mathcal{W}}_{ego}(i,k,j)\}_{j=1}^{n_c}$, $\mathcal{G}_r$ and $\mathcal{G}_s$, where n_c is the amount of disconnected items with u_i. For clarity, we simplify notation $\mathcal{G}_{ego}(i,k), \mathcal{W}_{ego}(i,k), \tilde{\mathcal{W}}_{ego}(i,k,j)$, $w_t(i,k)$ as $\mathcal{G}_{ego}, \mathcal{W}_{ego}, \tilde{\mathcal{W}}_{ego}(j), w_t$ in the following text respectively.

3.3 Friend Recommendation

Friend recommendation is critical for shaping and facilitating online social networks [33, 37], helping users to connect with each other. We adopt it as an auxiliary task in SoREX. We can formulate friend recommendation problem as follows:

Definition 5 (Friend recommendation): Given social network $\mathcal{G}_s$, the objective of friend recommendation is to predict missing links in $\mathcal{G}_s$, suggesting top-K disconnected users that the target users are most likely to make friend with.

Table 1. Summary of key notations used in SoREX.

(A) Data & Graph Structures	
$\mathcal{U}, \mathcal{V}$	Sets of users and items; $\lvert\mathcal{U}\rvert = m$, $\lvert\mathcal{V}\rvert = n$
$\mathcal{G}_r = (\mathcal{U} \cup \mathcal{V}, \mathcal{E}_r)$	User–item interaction bipartite graph
$\mathcal{G}_s = (\mathcal{U}, \mathcal{E}_s)$	User–user social graph
$\mathcal{G} = (\mathcal{U} \cup \mathcal{V}, \mathcal{E}_r \cup \mathcal{E}_s)$	Joint graph of $\mathcal{G}_r$ and $\mathcal{G}_s$
$\mathbf{R} \in \mathbb{R}^{m \times n}$	Adjacency / rating matrix of $\mathcal{G}_r$ ($\mathbf{R}_{ij} = 1$ if $(u_i, v_j) \in \mathcal{E}_r$)
$\mathbf{S} \in \mathbb{R}^{m \times m}$	Social adjacency matrix ($\mathbf{S}_{ij} = 1$ if $(u_i, u_j) \in \mathcal{E}_s$)
$\mathbf{A}^R$, $\mathbf{A}$	Block-adjacency matrices of $\mathcal{G}_r$ and $\mathcal{G}$ (Eq. 1)
$\mathcal{G}_{ego}(i, k)$	k-hop ego-net of user u_i
$\mathcal{W}_{ego}(i, k)$	Set of all ego-paths originated from u_i (Def. 3)
(B) Embeddings & Representations	
$\mathbf{E}^r$, $\mathbf{E}^s$	ID-embedding matrices for interaction tower / social tower
$\mathbf{e}^r_q$, $\mathbf{e}^s_q$	ID embedding of node q in the two towers
$\mathbf{h}^r_i$, $\mathbf{c}^r_j$	GNN-encoded user / item embeddings in interaction tower
$\mathbf{h}^s_i$	GNN-encoded user embedding in social tower
$\tilde{\mathbf{h}}^s_j$	Aggregated item representation in social tower (Eq. 11)
$\hat{\mathbf{h}}^r_i$, $\hat{\mathbf{h}}^s_i$	Explanation-re-aggregated final user embeddings (Eq. 16)
(C) Sampling & Explanation Variables	
w_t	One ego-path; $q \in w_t$ denotes nodes on the path
$\hat{\mathcal{W}}_{ego}$	Random-walk sampled path pool (size n_w)
$\tilde{\mathcal{W}}^r_{ego}(j)$, $\tilde{\mathcal{W}}^s_{ego}(j)$	Candidate-aware ego-path subsets in two towers
$p^{r*}_t(v_j)$, $p^{s*}_t(v_j)$	Raw path–item similarities (Eq. 10,11)
$p^r_t(v_j)$, $p^s_t(v_j)$	Normalised sampling probabilities (Eq. 13)
β^r_t, β^s_t	Bernoulli variables for path sampling
α^r_{qj}, α^s_{qj}	Attention weights in explanation re-aggregation
(D) Scores, Losses & Hyper-parameters	
$g_r(\cdot)$, $g_s(\cdot)$, $g(\cdot)$	Scoring functions of interaction tower, social tower, fused output
$f(u_i, u_q)$	Friend-recommendation score between users u_i and u_q
$\mathcal{L}_{main}$, $\mathcal{L}_s$	BPR loss for item ranking; BPR loss for friend recommendation
γ, λ	Loss-weight of auxiliary task; L_2-regularisation strength
k_1, k_2	GNN propagation layers in interaction / social tower
k	Hop number of ego-net / max walk length
n_w	Number of random walks sampled per user
d	Dimension of embeddings

4 Methodology

In this section, we present the methodology behind our proposed SoREX framework. We begin by providing an overview of SoREX and its architecture. Then, we explain the key components in detail, starting with the basic two-tower framework and moving on to the ego-path based explanation process, explanation re-aggregation, and multi-task training. Finally, we conclude with a computational complexity analysis.

4.1 Overview

The overview of our proposed SoREX is presented in Figure 3. As shown, SoREX adopts a two-tower architecture, equipped with two components: *ego-path sampling* and *explanation re-aggregation*, which aim to generate explanations

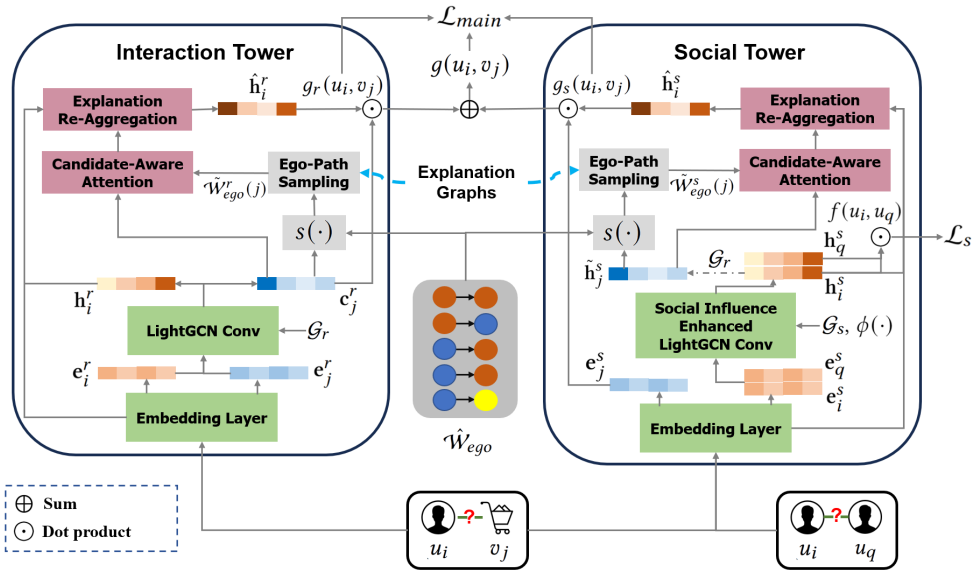


Fig. 3. Overview of our proposed SoREX framework for self-explainable GNN-based social recommendation.

and explicitly relate explanations to predictions respectively. Specifically, the basic two-tower framework is composed of an interaction tower and a social influence-aware social tower, laying the foundation for factor-specific explanations. Given a disconnected user-item pair and the user's ego-path set, each tower independently performs user/item embedding, ego-path sampling and item ranking, with final ranking obtained by fusing both tower predictions. In each tower, we compute the path-level similarity between each ego-path and candidate item, which is regarded as the seed of independent Bernoulli sampling. The sampled ego-paths are treated as candidate-aware explanations for further candidate-wise comparison, and they are further re-aggregated into the output user representations based on candidate-aware attention, so that the relevant ego-paths can be explicitly related to the downstream ranking predictions. To further enhance the model's performance and improve the user embeddings' ability to capture genuine relationships between friends in social graphs, multi-task learning is employed. The auxiliary task of friend recommendation helps the user embeddings better learn true friend relationships, compensating for the noise in the social graph. The notations used throughout the method are summarized in Table 1.

4.2 Basic Two-Tower Framework

To independently model social and user-item interaction factors and further lay the foundation for factor-specific explanation, we design a basic two-tower framework, which is composed of a social influence aware social tower and an interaction tower to learn from $\mathcal{G}_s$ and $\mathcal{G}_r$ respectively. Each tower is assigned with independent ID embeddings and GNN encoder, predicting its own ranking score. We define the ID embedding matrix in social and interaction tower as $\mathbf{E}^s \in \mathbb{R}^{(m+n)\times d}$ and $\mathbf{E}^r \in \mathbb{R}^{(m+n)\times d}$ respectively, where d is the embedding dimension. For interaction tower, we adopt k_1-layer LightGCN [11] to encode user/item representations for prediction:

$$\mathbf{h}_i^r = \frac{1}{k_1+1}[\mathbf{h}_i^{r^{(0)}} + \sum_{l=1}^{k_1} AGG^r(\mathbf{c}_j^{r^{(l)}} | j \in \mathcal{N}_i^r)], \tag{2}$$

$$\mathbf{c}_j^r = \frac{1}{k_1+1}[\mathbf{c}_j^{r^{(0)}} + \sum_{l=1}^{k_1} AGG^r(\mathbf{h}_i^{r^{(l)}} | i \in \mathcal{N}_j^r)], \tag{3}$$

$$AGG^r(\mathbf{x}_i | i \in \mathcal{N}_q^r) = \sum_{i \in \mathcal{N}_q^r} \frac{\mathbf{x}_i}{|\sqrt{\mathcal{N}_i^r}||\sqrt{\mathcal{N}_q^r}|}, \tag{4}$$

where $\mathbf{h}^r$ and $\mathbf{c}^r$ are the encoded user and item embeddings, $\mathbf{h}_i^{r^{(0)}}$ and $\mathbf{c}_j^{r^{(0)}}$ are assigned with corresponding ID embeddings $\mathbf{e}_i^r \in \mathbb{R}^d$ and $\mathbf{e}_j^r \in \mathbb{R}^d$ respectively. $\mathcal{N}_q^r$ is the set of direct neighbors of q on $\mathcal{G}_r$. The scoring function of interaction tower $g_r(u_i, v_j)$ is defined as the dot product of $\mathbf{h}_i^r$ and $\mathbf{c}_j^r$.

As for the social tower, we adopt a mutant of LightGCN as the GNN encoder. Empirical studies in [40] suggest that the influence of socially active users' purchase decisions on which of other users has no significant increase or decrease compared with inactive users. Thus, the neighborhood aggregation function in LightGCN is not suitable for user-item interaction modeling in social recommendation. As suggested in [19], socially connected users will have more influence on each other if they are of similar purchase behaviors. Therefore, we devise a social influence aware aggregation function and integrate it into original LightGCN architecture to encode $\mathcal{G}_s$. The derivation of encoded user embedding $\mathbf{h}^s$ is formulated as follows:

$$\mathbf{h}_i^s = \frac{1}{k_2+1}[\mathbf{h}_i^{s^{(0)}} + \sum_{l=1}^{k_2} AGG^s(\mathbf{h}_j^{s^{(l)}} | j \in \mathcal{N}_i^s)], \tag{5}$$

$$AGG^s(\mathbf{x}_i | i \in \mathcal{N}_q^s) = \sum_{i \in \mathcal{N}_q^s} \alpha_i \mathbf{x}_i, \tag{6}$$

$$\alpha_i = \underset{i \in \mathcal{N}_q^s}{\text{softmax}}\, \phi(i, q), \tag{7}$$

where k_2 is the number of convolutional layers in social tower, $\mathbf{h}_i^{s^{(0)}}$ is assigned with ID embedding $\mathbf{e}_i^s \in \mathbb{R}^d$, $\mathcal{N}_q^s$ is the set of direct neighbors of q on $\mathcal{G}_s$. $\phi(i, q)$ is the social influence function, evaluating the strength of social connection between u_i and u_q. We assume that a pair of socially connected users will have stronger social connections with each other if the overlap ratio of their interacted item sets is high. We could quantitatively measure such overlap with Jaccard similarity. However, we find that the difference of the Jaccard similarity among all available users is subtle. For simplicity, we adopt its square root to amplify the social influence difference. Therefore, $\phi(i, q)$ is formulated as follows:

$$\phi(i, q) = \sqrt{\frac{|\mathcal{N}_i^r \cap \mathcal{N}_q^r|}{|\mathcal{N}_i^r \cup \mathcal{N}_q^r|}}. \tag{8}$$

The scoring function of social tower $g_s(u_i, v_j)$ is defined as the dot product of $\mathbf{h}_i^s$ and the ID embedding of candidate item $\mathbf{e}_j^s$. With g_s and g_r, the final scoring function g of the basic framework is formulated as:

$$g(u_i, v_j) = g_r(u_i, v_j) + g_s(u_i, v_j). \tag{9}$$

Note that the basic framework is flexible for that the encoder in each tower could be replaced by any advanced GNN encoders. As for the potential impact, we leave it for future work.

4.3 Ego-Path Based Explanation

This section presents the explanation approach used in SoREX. We aim to provide substructure-based explanations to enhance transparency of social recommenders. Previous explainable recommendation methods for transparency enhancement fail to utilize pure structural information.

Although existing self-explainable GNNs are able to investigate complex explanatory substructures, they fail to provide comparative explanations for ranking predictions. To address these challenges, we propose to sample relevant subgraphs of $\mathcal{G}_{ego}$ for each candidate item v_j and each concerned factor as candidate-aware and factor-specific explanation graph. This process is equivalent to sample subsets of $\mathcal{W}_{ego}$. Considering that the size of $\mathcal{W}_{ego}$ grows exponentially with the increase of hop number k, it is not feasible to perform path ranking over the whole $\mathcal{W}_{ego}$. Inspired by [56], we randomly and uniformly sample n_w walks originated from target user u_i with maximum length k. Then we transform the sampled paths into the form of ego-paths introduced in section 3. This procedure essentially selects an ego-path subset $\hat{\mathcal{W}}_{ego}$ from all available ego-paths in $\mathcal{W}_{ego}$ to extract an even smaller explanation graph from. Note that repetitive walks will not be removed from $\hat{\mathcal{W}}_{ego}$ in order to implicitly reflect structural features of $\mathcal{G}_{ego}$.

Given candidate item v_j, the sampling probability for ego-path $w_t \in \hat{\mathcal{W}}_{ego}$ is determined by similarity between w_t and v_j in specific tower. We take the average similarity between candidate v_j and each node q in w_t as the path-level similarity. We formulate $p_t^{r*}(v_j)$ and $p_t^{s*}(v_j)$, the sampling seed in interaction tower and social tower for w_t in Eq. (10) and Eq. (11) respectively:

$$p_t^{r*}(v_j) = \frac{1}{k-1}\Big(\sum_{q \in \mathcal{U} \cap w_t} s(\mathbf{h}_q^r, \mathbf{c}_j^r) + \sum_{q \in \mathcal{V} \cap w_t} s(\mathbf{c}_q^r, \mathbf{c}_j^r)\Big), \tag{10}$$

$$p_t^{s*}(v_j) = \frac{1}{k-1}\Big(\sum_{q \in \mathcal{U} \cap w_t} s(\mathbf{h}_q^s, \tilde{\mathbf{h}}_j^s) + \sum_{q \in \mathcal{V} \cap w_t} s(\tilde{\mathbf{h}}_q^s, \tilde{\mathbf{h}}_j^s)\Big), \tag{11}$$

$$\tilde{\mathbf{h}}_j^s = \begin{cases} \dfrac{1}{|\mathcal{N}_j^r|} \displaystyle\sum_{i \in \mathcal{N}_j^r} \mathbf{h}_i^s, & |\mathcal{N}_j^r| > 0 \\ \mathbf{e}_j^s, & |\mathcal{N}_j^r| = 0 \end{cases}, \tag{12}$$

where $s(\cdot)$ is cosine similarity. The probability computation in interaction tower is directly based on the output representations of GNN encoder. In Eq. (10) and Eq. (11), the denominator is $k-1$ because we are excluding the target user when computing the path-level similarity. The value of $k-1$ corresponds to the length of the path excluding the starting node, as we are only interested in the relationships between the target user and other nodes in the path. In social tower, the user representations used for probability computation are also the output of GNN encoder. Considering that item nodes do not participate in message propagation in $\mathcal{G}_s$, to fully leverage structural knowledge in social GNN encoder, we take the mean pooling of the GNN-encoded representations of users that have interacted with candidate v_j as item representation used for probability computation, which corresponds to $\tilde{\mathbf{h}}_j^s$. If v_j is a cold-start item with no interactions, we will replace $\tilde{\mathbf{h}}_j^s$ with v_j's ID embedding. We further adjust $p_t^{r*}(v_j)$ and $p_t^{s*}(v_j)$ into range $[0, 1]$ as sampling probabilities $p_t^r(v_j)$ and $p_t^s(v_j)$:

$$p_t^r(v_j) = \frac{p_t^{r*}(v_j) + 1}{2}, \quad p_t^s(v_j) = \frac{p_t^{s*}(v_j) + 1}{2}. \tag{13}$$

Then, we sample ego-path subsets $\tilde{\mathcal{W}}_{ego}^r(j)$ and $\tilde{\mathcal{W}}_{ego}^s(j)$ based on Bernoulli distribution $\beta_t^r \sim \text{Bern}(p_t^r(v_j))$ and $\beta_t^s \sim \text{Bern}(p_t^s(v_j))$ in interaction tower and social tower respectively. We only keep ego-paths with $\beta_t^* = 1$ in corresponding

tower. To make sure the gradient w.r.t. $p_t^r(v_j)$ and $p_t^s(v_j)$ is computable, we apply gumble-softmax reparameterization trick [17]. To this end, we have obtained two ego-path subsets relevant to v_j from the view of interaction and social factor respectively. They will be regarded as the factor-specific explanation graphs for v_j's ranking result.

Our ego-path based explanations can effectively address aforementioned challenges. The sampled ego-paths can form dense explanation graphs. Ideally, ego-paths interweave with each other and form various motifs, enabling investigation of complex substructures based on their quantity and importance. Besides, we can easily compare different candidate-aware and factor-specific ego-path subsets and come up with comparative explanations from different perspectives, filling the gap of comparative explainability in GNN-based social recommenders.

4.4 Explanation Re-Aggregation

To make SoREX self-explainable, we need to relate explanation to predictions. Many self-explainable methods conduct downstream prediction directly based on explanations. To reduce information loss, we instead treat the sampled ego-path subsets as an emphasis of the relevant part of $\mathcal{G}_{ego}$ and perform explanation re-aggregation to relate the sampled ego-paths with final prediction.

Explanation re-aggregation aims to aggregate information from sampled ego-paths into original GNN-encoded representations of target users in corresponding tower, such that the factor-specific and candidate-aware knowledge within $\mathcal{G}_{ego}$ can be emphasized. A hop-wise attention based node-level aggregation method is designed for this procedure. We take interaction tower as an example for illustration. Given GNN-encoded user embedding $\mathbf{h}_i^r$, candidate item v_j and interaction-specific v_j-aware ego-path subset $\tilde{\mathcal{W}}_{ego}^r(j)$, we first compute attention in the fashion of Transformer [41] for each node q in $w_t \in \tilde{\mathcal{W}}_{ego}^r(j)$ based on its cosine similarity with the GNN-encoded representation of v_j:

$$a_{qj}^r = \begin{cases} \dfrac{s(\mathbf{h}_q^r, \mathbf{c}_j^r)}{\sqrt{d}}, q \in \mathcal{U} \\ \dfrac{s(\mathbf{c}_q^r, \mathbf{c}_j^r)}{\sqrt{d}}, q \in \mathcal{V} \end{cases}. \tag{14}$$

We use $\tilde{\mathcal{W}}_{ego}^r(j)[l], l \in \{1, 2, ..., k\}$ denote the collection of the l-th hop node in $\forall w_t \in \tilde{\mathcal{W}}_{ego}^r(j)$. For node q in the l-th hop, its normalized attention value α_{qj}^r is formulated as follows:

$$\alpha_{qj}^r = \underset{q \in \tilde{\mathcal{W}}_{ego}^r(j)[l]}{\text{softmax}} a_{qj}^r. \tag{15}$$

After hop-wise attention normalization, we perform ego-path aggregation via simple addition based on the ID embeddings of nodes involved in sampled ego-paths. The interaction-specific explanation enhanced final user embedding is defined as Eq. (16):

$$\hat{\mathbf{h}}_i^r = \frac{1}{k+1}(\mathbf{h}_i^r + \sum_{w_t \in \tilde{\mathcal{W}}_{ego}^r(j)} \sum_{q \in w_t} \alpha_{qj}^r \mathbf{e}_q^r). \tag{16}$$

The attention weights α_{qj}^s and explanation enhanced final user embedding $\hat{\mathbf{h}}_i^s$ in social tower can be similarly defined with $\mathbf{h}_i^s, \tilde{\mathbf{h}}_j^s$ and $\mathbf{E}^s$. Given $\hat{\mathbf{h}}_i^s$ and $\hat{\mathbf{h}}_i^r$, we redefine $g_r(u_i, v_j)$ as the dot product of $\hat{\mathbf{h}}_i^r$ and $\mathbf{c}_j^r$, and redefine $g_s(u_i, v_j)$ as dot product of $\hat{\mathbf{h}}_i^s$ and $\mathbf{e}_j^s$. In Eq. (16), the denominator is $k+1$ because we include the target user in the aggregation process. This ensures that the representation of the user is influenced by both their original embedding and the embeddings of the nodes in the sampled ego-paths, including the starting node.

4.5 Multi-Task Training

We employ multi-task learning to optimize two tasks jointly in SoREX. The user-item interaction modeling task serves as the major task, and we adopt BPR loss for pairwise ranking. We conduct optimization for each tower's own predictions and the fused final predictions. The loss function $\mathcal{L}_{main}$ for the primary task is formulated as follows:

$$\mathcal{L}_{main} = \sum_{(u,v,v^-)\in\mathcal{E}_t} -\log\sigma(g_r(u,v) - g_r(u,v^-)) - \log\sigma(g_s(u,v) - g_s(u,v^-)) - \log\sigma(g(u,v) - g(u,v^-)), \tag{17}$$

where v and v^- represent positive and negative sample respectively, and $\mathcal{E}_t$ is the training set. The other auxiliary task is friend recommendation task. The introduction of friend recommendation can help distill more supervision signals from social domain in the view of user-user relationships, instead of the relationships between social proximity and user-item interactions only. Recent work [14] has demonstrated its positive effect for social recommendation. Therefore, we randomly sample positive and negative user pairs from $\mathcal{G}_s$, and conduct candidate user ranking and item ranking simultaneously. Considering that friend recommendation only leverages social networks, we only integrate this task into social tower. The loss function of friend recommendation $\mathcal{L}_s$ is defined as:

$$f(u_i, u_q) = \mathbf{h}_i^s \cdot \mathbf{h}_q^{s^T}, \tag{18}$$

$$\mathcal{L}_s = \sum_{(u,u^+,u^-)\in\mathcal{E}_s} -\log\sigma(f(u,u^+) - f(u,u^-)), \tag{19}$$

where f is the scoring function for user pairs. The overall objective function $\mathcal{L}$ is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{main} + \gamma\mathcal{L}_s + \lambda(\| \mathbf{E}^s \|_2 + \| \mathbf{E}^r \|_2), \tag{20}$$

where the last term is the L2 regularization, and γ, λ are hyperparameters to control the strength of regularization. Note that the only trainable parameters in SoREX are $\mathbf{E}^s$ and $\mathbf{E}^r$.

4.6 Computational Complexity

This subsection aims to analyze the time and memory complexity of SoREX. Considering that representative baseline methods (e.g. LightGCN [11], Diffnet [48] and DESIGN [40]) essentially adopt different graph convolution strategies over the same joint graph $\mathcal{G}$, we compare our SoREX with the more computationally efficient method LightGCN and the multi-tower alike method DESIGN. The time and memory complexity of SoREX and the selected baselines are listed in Table 2. Suppose that $|\mathcal{U}|$ and $|\mathcal{V}|$ are the size of user set and item set, while $|\mathcal{E}|$ is the size of edge set of $\mathcal{G}$. Given the dimension of learnable embeddings d and the layer number L, the time complexity of LightGCN and SoREX is $O(L|\mathcal{E}|d)$ and $O(L|\mathcal{E}|d + n_w|\mathcal{E}_r|d)$ respectively. DESIGN essentially performs message propagation over $\mathcal{G}$ for twice, so the time complexity of DESIGN is also $O(L|\mathcal{E}|d)$. The optimal n_w usually satisfies $L \ll n_w \ll |\mathcal{E}|$. Therefore, the time complexity of SoREX is one to two orders of magnitude greater than that of LightGCN and DESIGN, which is acceptable when dealing with sparse networks. On the other hand, SoREX has no other trainable parameters except for the ID embeddings, which is same as LightGCN and DESIGN. Considering that the ego-path sampling procedures are real-time, the memory costs of SoREX is nearly the twice of LightGCN. Therefore, the memory complexity of SoREX and LightGCN are both $O(L(|\mathcal{U}| + |\mathcal{V}|)d)$. Items are not assigned with ID embeddings in DESIGN, so the memory complexity of DESIGN is $O(L|\mathcal{U}|d)$. In summary, although the time and memory complexity of SoREX are not superior,

we aim to achieve the necessary trade-off between efficiency and explainability, while minimizing the unnecessary computational costs.

Table 2. Computational Complexity Comparison.

	Time	Memory
LightGCN	$O(L\|\mathcal{E}\|d)$	$O(L(\|\mathcal{U}\| + \|\mathcal{V}\|)d)$
DESIGN	$O(L\|\mathcal{E}\|d)$	$O(L\|\mathcal{U}\|d)$
SoREX	$O(L\|\mathcal{E}\|d + n_w\|\mathcal{E}_r\|d)$	$O(L(\|\mathcal{U}\| + \|\mathcal{V}\|)d)$

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate our SoREX on several widely used benchmark datasets, including three sparse datasets *Yelp*[1], *Flickr*[2], *Ciao*[3] and a dense dataset *LastFM*[4]. Yelp is an online location-based social network. Flickr is an online image-based social sharing platform with a whom-trust-whom social network. Ciao is a popular social networking website. LastFM consists of a user-artist interaction network and a user friendship network. Consistent with previous work [48], we remove users and items with less than two interaction records. For datasets with ratings like Ciao, we only keep links with ratings no less than 4. Considering that all the selected datasets have no temporal information, we randomly split each dataset into train/validation/test sets at a ratio of 80%/10%/10%. Detailed preprocessed dataset statistics are presented in Table 3.

Table 3. Preprocessed Dataset Statistics.

Dataset	#User	#Item	#Interaction	Density	#Social
Yelp	17,220	35,351	205,529	0.034%	143,609
Flickr	8,137	76,190	320,775	0.052%	182,078
Ciao	6,788	77,248	206,143	0.039%	110,383
LastFM	1,880	3,933	75,228	1.017%	25,260

Baseline Methods. To validate the effectiveness of SoREX, we select two groups of baselines for comparison:
(1) GNN-based social recommendation baselines:

- *LightGCN* [11] is a popular GNN model for general recommendation task. It is characterized by recursive graph convolution without linear transformation and non-linear activation function.
- *LightGCN*^{S} is the base model of CGCL [14], which is a friend recommendation enhanced LightGCN, conducting additional message propagation on social network to predict social links.
- *SocialLGN* [26] performs message propagation on both $\mathcal{G}_r$ and $\mathcal{G}_s$ in each GNN layer to fuse their knowledge.
- *Diffnet* [48] is a layer-wise social influence propagation model proposed to simulate the social influence diffusion process.
- *Diffnet++* [47] is an extension of Diffnet, modeling both social influence diffusion and user preference diffusion in a unified framework.

[1]https://github.com/librahu/HIN-Datasets-for-Recommendation-and-Network-Embedding
[2]https://www.flickr.com/
[3]https://www.cse.msu.edu/tangjili/datasetcode/truststudy.htm
[4]https://files.grouplens.org/datasets/hetrec2011

- *MHCN* [61] is a multi-channel hypergraph convolutional network. We remove the self-supervised learning related components in MHCN and test the effectiveness of its proposed model architecture.
- *DESIGN* [40] leverages multiple GNN encoders to learn from different graphs and fuse their knowledge via knowledge distillation.
- *GBSR* [54] leverages graph information bottleneck theory to identify and remove redundant social links, so that minimal but sufficient social network could be used for enhanced social recommendation.

(2) Self-explainable GNN or recommendation baselines:

- *ConPI* [45] compares the similarity between neighbor sets of node pairs for both link prediction and explanation. Two versions of ConPI are devised. ConPI-Node identifies similar neighbors as explanations, while ConPI-Pair leverages similar neighbor pairs. We only test ConPI-Node here for the excessive memory complexity of ConPI-Pair.
- *ECF* [6] is a self-explainable collaborative filter method, which aims to discover taste clusters from user-item interactions. The approximation relationships between learned taste clusters and items are used for explanation. Item tags are not adopted for cluster description in our experiment, while only user-item interaction links are used in the evaluation of ECF.
- *GSAT* [30] is a self-explainable attention-based GNN proposed to inject learnable randomness into GNN with information bottleneck theory and sample dense explanatory subgraphs without spurious correlations.
- *PxGNN* [4] is a prototype-based self-explainable GNN, which aims to generate prototype graphs for both prediction and prototype-based explanations. To transplant PxGNN into link prediction task, we set the amount of node class as 1, so that all the nodes share the same set of prototype graphs.

Note that both ConPI and GSAT are trained on the joint graph $\mathcal{G}$. We do not adopt other representative self-explainable GNN baselines like SE-GNN [3], ProtGNN [67], PGIB [31] and DisC [7] for comparison, because they are specifically designed for classification tasks. It is infeasible to extract node pair aware explanatory subgraphs via these methods, and thus it is non-trivial to directly transfer them to our task. We also exclude social recommendation baselines SEPT [59], DcRec [46] and CGCL [14], which focus on self-supervised learning (SSL) for social recommenders. For recent social network denoising based methods [21, 25, 54], we only select GBSR for evaluation because their underlying ideas are similar. We would like to compare the superiority of different methods from the perspective of model architecture, so we exclude the SSL related components in the three baselines and test their underlying models, which correspond to LightGCN and LightGCN^S.

Evaluation Metrics. Following [40], we adopt Hit Rate (HR@K) and Normalized Discounted Cumulative Gain (NDCG@K) as evaluation metrics. HR@K evaluates the accuracy of recommendation, while NDCG@K can reflect the quality of item ranking. We set $K = 10$ for both HR@K and NDCG@K.

Implementation Details. We implement our framework and all selected baselines with PyTorch-Geometric [9]. We fix the ID embedding dimension size d as 64 for all methods. For SoREX, hop number of ego-paths k, layer number of GNN encoders k_1 and k_2 are all tuned from $\{1, 2, 3\}$. n_w, the size of $\hat{\mathcal{W}}_{ego}$, is tuned from $\{50, 100, 150, 200, 300, 500\}$. The friend recommendation task coefficient γ is tuned from $\{0.1, 0.2, ..., 1.0\}$. For LightGCN and LightGCN^S, we adopt 2-layer GNN architecture. The coefficient of auxiliary task in LightGCN^S is tuned from the same range as γ. For all other baselines, we adopt the recommended settings in their paper. Adam optimizer [20] is adopted for the optimization of all models with learning rate 0.001. The batch size is tuned from $\{256, 512, 1024\}$, and L2 regularization coefficient λ is set as 0.001. Following [48], for all methods, we randomly sample 10 and 1000 negative samples for each training and

validating sample respectively. During testing, we rank all items that target users have not interacted with. Our code and data are available at https://github.com/antman9914/SoREX.

Table 4. Overall performance. Boldfaced and underlined scores are the best and the second-best ones respectively. The improvements achieved by SoREX are statistically significant (p-value $\ll 0.05$).

	Yelp		Flickr		Ciao		LastFM	
	HR@10	NDCG	HR@10	NDCG	HR@10	NDCG	HR@10	NDCG
LightGCN	0.0192	0.0089	<u>0.0045</u>	<u>0.0023</u>	0.0377	0.0205	0.1875	0.1042
LightGCN^S	<u>0.0215</u>	<u>0.0100</u>	0.0043	0.0021	<u>0.0384</u>	<u>0.0212</u>	0.1884	<u>0.1064</u>
SocialLGN	0.0179	0.0086	0.0040	0.0020	0.0355	0.0180	0.1650	0.0941
Diffnet	0.0187	0.0083	0.0026	0.0012	0.0224	0.0096	0.1497	0.0822
Diffnet++	0.0155	0.0076	0.0031	0.0014	0.0267	0.0132	0.1767	0.0987
MHCN	0.0183	0.0094	0.0044	0.0021	0.0326	0.0162	<u>0.1931</u>	0.1029
GBSR	0.0206	0.0097	0.0042	0.0021	0.0363	0.0182	0.1906	0.1045
DESIGN	0.0199	0.0091	0.0038	0.0019	0.0344	0.0187	0.1577	0.0904
ConPI	0.0053	0.0024	0.0010	0.0005	0.0051	0.0025	0.0691	0.0294
ECF	0.0174	0.0082	0.0038	0.0017	0.0345	0.0177	0.1635	0.0952
GSAT	0.0197	0.0087	0.0040	0.0021	0.0324	0.0167	0.1810	0.1003
PxGNN	0.0185	0.0080	0.0034	0.0015	0.0308	0.0149	0.1664	0.0985
SoREX	**0.0227**	**0.0104**	**0.0047**	**0.0024**	**0.0402**	**0.0221**	**0.1998**	**0.1144**
Improvement	5.58%	4.00%	4.44%	4.35%	4.69%	4.25%	3.47%	7.52%

5.2 Main Results

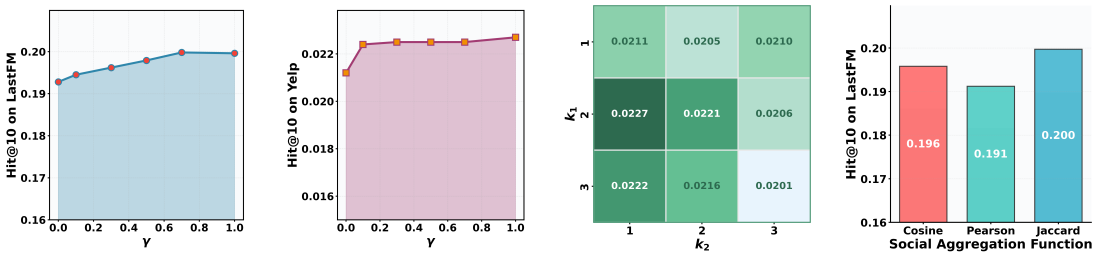
We report the average evaluation results of 5 runs with different random seeds in Table 4, where NDCG is short for NDCG@10 results. Due to the existence of randomness in SoREX and GSAT, we take the average performance of 5 times of tests for each run. The relative improvement is also presented. We have the following observations: (1) Our SoREX significantly outperforms other baseline methods on all metrics and all datasets, which demonstrates the superiority of our proposed model architecture. (2) LightGCN and LightGCN^S are the most competitive baselines, which demonstrates the superiority of pure message propagation structure in social recommendation. (3) The performance of baselines specifically designed for social recommendation significantly depends on dataset. (4) The performance of ConPI, GSAT and PxGNN are inferior in social recommendation, which could be the result of inappropriate underlying model architecture.

Compared with the selected baselines, we attribute the success of SoREX to following designs: (1) an independent two-tower architecture, which enables independent modeling of both social and user-item interaction factors. (2) explanation re-aggregation, which emphasizes the relevant subgraphs of target users' ego-nets and assists the final predictions. (3) The integration of friend recommendation task, whose effectiveness is also verified in LightGCN^S.

5.3 Hyperparameter Analysis

To enable a more in-depth empirical analysis, we conduct a hyperparameter study focusing on three key aspects: (1) the multi-task balancing parameter γ, (2) the number of layers k_1 and k_2 in the social graph and the recommendation graph, respectively, and (3) the choice of the social aggregation function.

As shown in Figures 4(a) and 4(b), the model performs relatively poorly when γ is set to 0. Performance stabilizes when γ ranges between 0.1 and 1.0, underscoring the importance of the multi-task design in enhancing social recommendation.



(a) (b) (c) (d)

Fig. 4. Hyperparameter sensitivity analysis. The figure presents the effects of varying the key hyperparameters on the model's performance, including the choice of the aggregation function, the multi-task balancing parameter γ, and the number of layers k_1 and k_2 in both graphs. The rightmost bar (*Jaccard*) is significantly better than Cosine and Pearson at the 5% level (paired t-test, $p < 0.05$).

Figure 4(c) illustrates the effect of k_1 and k_2 on Hit@10 performance using the Yelp dataset. Notably, the social GNN (k_2) achieves better results with fewer layers (1–2), likely due to the higher noise level in social graphs compared to user–item interaction graphs.

In our framework, social influence is modeled using Jaccard similarity, which captures the proportion of shared neighbors between users. To assess the effectiveness of this design, we compare Jaccard with two commonly used alternatives: cosine similarity and the Pearson correlation coefficient. As shown in Fig. 4(d), the Jaccard-based strategy consistently outperforms both. This superior performance stems from Jaccard's ability to directly reflect discrete overlap in social connections—an especially meaningful signal in social networks, where shared neighbors often imply stronger influence and homophily. In contrast, cosine similarity relies on the orientation of user embedding vectors, making it more susceptible to noise, while Pearson correlation is tailored for continuous variables and performs poorly with binary relational data such as friendship links (1 or 0). These results confirm that Jaccard similarity is more suitable for modeling social influence in our setting.

5.4 Ablation Study

To demonstrate the effectiveness of our proposed components, we conduct two series of ablation experiments on Yelp dataset: *(1) General ablation studies:* We first remove auxiliary objective function $\mathcal{L}_s$ (w/o $\mathcal{L}_s$), our proposed explanation re-aggregation procedure (w/o re-aggr) and all components related to social tower (w/o soc tower) respectively. Then, we replace the social influence aware LightGCN encoder in social tower with the original LightGCN (w/o soc impact). *(2) Alternative design studies:* We first attempt to replace the ego-path sampling procedure with top-K ego-path extraction to generate sparse explanation graph (w/ top-K ego-path). Then, referring to [30], we test the effect of additional information bottleneck (IB) based constraints on the sampling probability distributions in both towers, trying to remove potential spurious correlations in explanatory subgraphs (w/ IB constraint). We also try to share the same ID embeddings between the two towers (w/ shared ID), and replace user-transformed item representations $\tilde{h}^s_j$ used for sampling probability computation in social tower with their corresponding ID embeddings e^s_j (w/o trans item emb).

Based on the results in Table 5, we can find that the full version of SoREX outperforms all other variants. We have following observations: *For verification of the proposed components:* (1) Both social context and friend recommendation auxiliary task are significantly beneficial for recommendation. Their removal both result in performance drop of approximately 10%. (2) The removal of social influence modeling causes significant performance drop, which demonstrates

Table 5. Ablation study on Yelp.

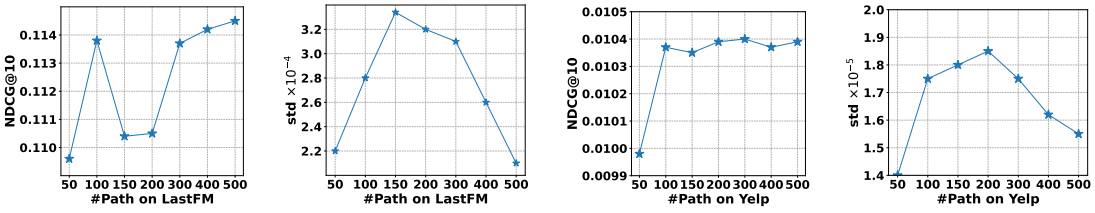
	HR@10	MRR	NDCG
SoREX	**0.0227**	**0.0113**	**0.0104**
w/o $\mathcal{L}_s$	0.0207	0.0103	0.0091
w/o re-aggr	0.0215	0.0109	0.0100
w/o soc impact	0.0209	0.0106	0.0095
w/o soc tower	0.0205	0.0101	0.0092
w/ top-K ego-path	0.0200	0.0100	0.0091
w/ IB constraint	0.0217	0.0110	0.0100
w/ shared ID	0.0209	0.0109	0.0098
w/o trans item emb	0.0210	0.0106	0.0095

the effectiveness of user preference based social influence. (3) Utilizing information from explanation graphs can improve predictive accuracy by 4-5%, indicating the necessity for emphasizing relevant neighborhood of target user. *For alternative designs:* (1)The variant using sparse explanation graphs is inferior than the full SoREX, because dense graphs can keep ego-paths that are less relevant but important for item ranking. (2) Integration of IB based constraint does not improve accuracy, indicating that the constraint is not suitable for our proposed attention mechanism. (3) Based on the performance drop in variant "w/ shared ID", independent modeling of social and interaction information is necessary. (4) User-transformed item representation is helpful for fully utilization of structural knowledge within social GNN encoder.

5.5 Stability Analysis

Ego-path generation in SoREX relies on random walk sampling, which introduces inherent randomness that may affect prediction stability—particularly in dense networks or for high-degree nodes. To examine this effect, we conduct experiments on two representative datasets: LastFM (a relatively dense dataset with density >1%) and Yelp (the sparsest among the four, with a density of 0.034%). Comparing these datasets allows us to evaluate SoREX under different levels of sparsity: in sparse graphs like Yelp, lower node degrees and smaller path selection spaces naturally reduce sampling noise, leading to more stable performance. This dataset-specific analysis provides deeper insight into SoREX's robustness across different scenarios.

In our experiments, we fix the random seed and retrain SoREX multiple times, varying the number of sampled ego-paths. For each path quantity, we perform 5 test runs and report the average NDCG@10 and standard deviation in Figure 5 to analyze sensitivity. Our observations are as follows: (1) Incorporating ego-paths improves recommendation performance, yet simply increasing their number does not always yield further gains, as shown in Figure 5(a) and Figure 5(c). (2) While incorporating more ego-paths generally improves stability, this holds true only after reaching a minimum threshold. Below this threshold, adding paths may increase variance, likely due to insufficient sampling to represent the ego-network's statistical properties. (3) Despite minor fluctuations in standard deviation with varying ego-path quantities, the performance oscillation on LastFM remains within a narrow range of 0.2–0.3%(Figure 5(b)), indicating high robustness. (4) As shown in Figure 5(d), SoREX achieves even more stable results on the sparser Yelp dataset. The smaller node degrees and reduced path selection space lower the potential noise, allowing the model to reach optimal performance with fewer paths.



(a) (b) (c) (d)

Fig. 5. Stability analysis of SoREX with respect to the number of ego-paths on LastFM and Yelp. (a) and (b) report the average and standard deviation of NDCG@10 on LastFM as the number of ego-paths varies. (c) and (d) show the corresponding results for Yelp.

5.6 Explanation Analysis

We analyze our generated explanations from both qualitative and quantitative aspects. We first conduct case study for instance-level qualitative analysis to further illustrate the idea of comparative explanation, and then provide systematic pattern analysis to demonstrate SoREX's ability to investigate important substructures for recommendation. We also conduct quantitative analysis based on fidelity score [55]. All discussions are based on the setting $k = 2$. Only testing samples with ground truth items ranked within top-5 are considered.

Case Study. We conduct a case study to analyze the relations between explanatory subgraphs and ranking decisions. We select a representative test sample from Yelp dataset, visualizing the ego-paths activated by the positive sample and one of the low-ranking negative samples in each tower in Figure 6. We also visualize the 1-hop ego-nets of involved items to show the relations between candidates and candidate-aware explanations. We remove the repetitive ego-paths and reconnect ego-paths with the target user to be more intuitive. Following observations can be concluded: (1) Different towers have different focuses. Based on explanations for positive sample in Figure 6(a) and 6(b), we can find that the sampled ego-paths and their similarity distributions in different towers are quite different. In interaction tower, only a few ego-paths are of high similarity, and social relation based common neighborhood is paid more attention to. In contrast, more ego-paths are of high similarity in social tower, and more user-item interaction based common neighborhood is identified, which indicates that the user-item pair is closer from social perspective. (2) Items that have no common neighborhood with target user can also activate certain ego-paths as explanations. Based on negative sample related explanations in Figure 6(c) and 6(d), although there is no overlap between the neighborhood of negative sample and target user, we can still identify important ego-paths. (3) Comparative explanation can be made by directly comparing explanations of different samples. Based on the presented example, several paths can be found between target user and positive sample. The ego-paths are also of higher similarity with positive sample in both towers. In contrast, the selected ego-paths are less relevant with negative sample, and there is no common neighborhood between negative sample and target user. To this end, we have provided reasonable explanations for ranking the positive sample higher than the negative sample via comparison among explanation graphs.

Note that although the dense explanatory graphs make them not easily understandable for end users, our design aims to investigate comparative relationships. Dense explanatory graphs can provide detailed candidate-wise comparison, which can be more persuasive for both users and developers after further processing.

Systematic Pattern Analysis. With ego-paths intertwined, various motifs can be formed. To demonstrate SoREX's ability for model-level explanation, we provide a toy example and analyze the systematic patterns of two simple motifs

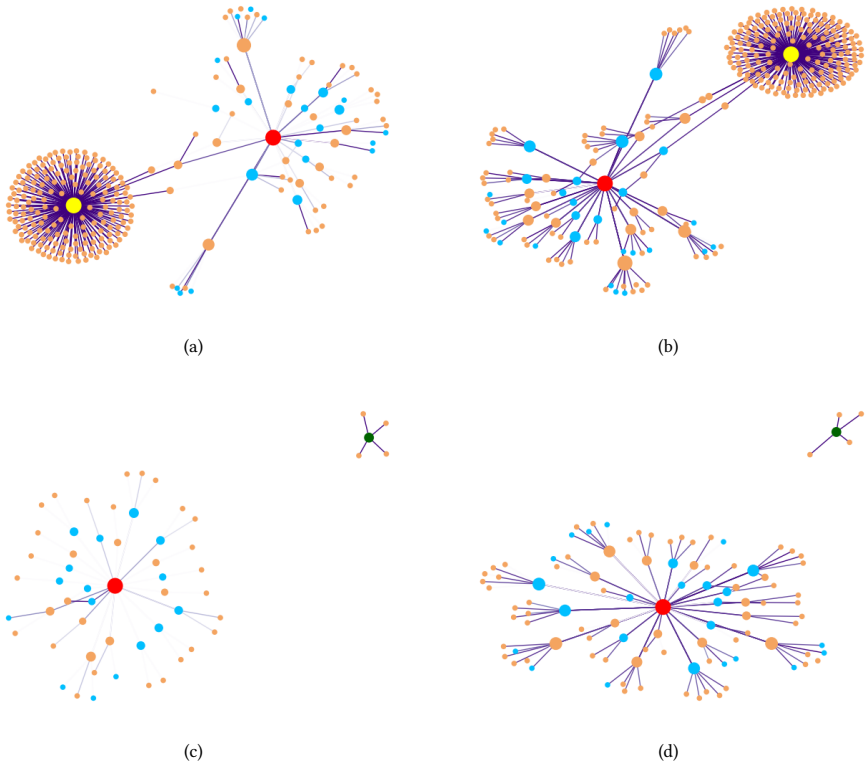


Fig. 6. Explanation graph visualization for case study. (a) and (b) are explanation graphs for positive sample in interaction tower and social tower respectively, while (c) and (d) are explanations for the low-ranking negative sample in the same tower order. For all four graphs, the orange and blue nodes represent user and item nodes, while the red, yellow and blackish green nodes represent target user, positive sample and negative sample respectively. Note that the blackish green negative sample nodes are in the upper-right corner of (c) and (d). The shade of edges in the user's ego-net represent the cosine similarity between their targeting nodes and corresponding candidate item.

formed when two ego-paths intersect, namely triangles and quadrilaterals. Based on the number of user and item nodes in motifs, we can categorize them into two types of triangles and three types of quadrilaterals. Figure 7 presents templates and the real-world meanings of these detailed motif types. Considering that quadrilaterals are essentially two ego-paths with the same endpoints, we analyze quadrilaterals based on the types of ego-paths alternatively for simplicity. There are totally three kinds of templates for ego-paths, including "user-user-user" (friend of friend, "fof" for short), "user-item-user" (co-purchase, "cop" for short) and "user-user-item" (friend's interaction, "fi" for short). We leave more complex substructures for future work. Our discussions will be extended from the perspective of detection rate and average similarity derived from Eq. (13), which represent the quantity and importance of motifs respectively. Comparisons will also be made between two towers and between positive and negative samples to identify factor-specific and comparative patterns. Note that detection rate and average similarity relate to each other, because the similarity values directly determine the sampling probability.

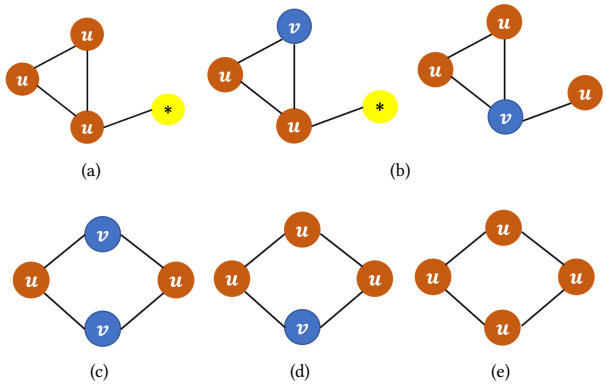
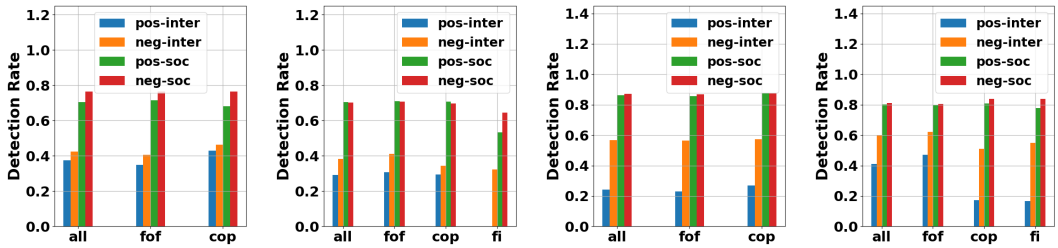
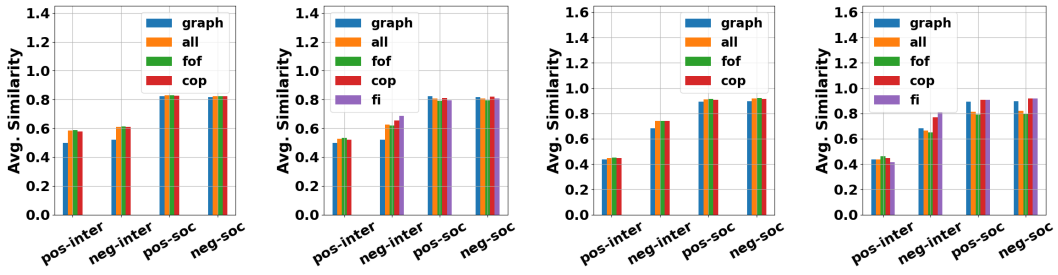


Fig. 7. Possible templates of triangles and quadrilaterals. (a)-(b) are two triangle types representing "friend of friend" ("fof" for short) relations in social network and co-purchase ("cop" for short) relations in interaction graph respectively. (c)-(e) are three quadrilateral types representing different combinations of "cop" and "fof" relations, indicating stronger connections among endpoint users.



(a) Triangles in Yelp (b) Quads in Yelp (c) Triangles in Ciao (d) Quads in Ciao

Fig. 8. Detection rate of triangles and quadrilaterals (quads for short) on Yelp and Ciao. The results are grouped by motif types, where "all" refers to motif-level statistics. Detailed data of different towers and positive/negative samples is also distinguished. Note that the fi motif in (b) is omitted, as there are no positive interactions for this type in the dataset.



(a) Triangles in Yelp (b) Quads in Yelp (c) Triangles in Ciao (d) Quads in Ciao

Fig. 9. Average similarity of triangles and quadrilaterals (quads for short) on Yelp and Ciao. The results are grouped by their tower and positive/negative sample belongings. Detailed data of different types of motifs is also presented.

We first analyze the detection rate of both triangles and quadrilaterals. Given motif type, detection rate measures the ratio of detected motifs to all motifs formed in $\hat{\mathcal{W}}_{ego}$. The motif-level and type-specific data on Yelp and Ciao are shown in Figure 8. We have following observations: (1) In both datasets, detection rate of both triangles and quadrilaterals for positive sample related explanation graphs are less than the rate for negative samples related explanations. Considering that positive samples usually have common neighborhood with target users, the ego-paths connecting them will grab more attention when processing positive samples, while motifs with relational information will be more important for negative samples without common neighborhood. (2) Detection rate in social tower is generally higher than which in interaction tower. (3) The detection rate of triangles is higher than quadrilaterals in Yelp, while the conclusion is opposite in Ciao. (4) In both datasets, interaction based "cop" triangles and social based "fof" quadrilaterals have higher detection rate in interaction tower for explanations of both positive and negative samples. There are no significant patterns in social tower.

Then we analyze the average similarities. We use the average similarity of all ego-paths involved in a motif as its similarity. The tower-specific data of Yelp and Ciao are shown in Figure 9. Following observations can be concluded: (1) In both datasets, the average similarities of all types of triangles are consistently higher than which of all ego-paths in both towers, indicating that triangles are important motifs in explanations. In contrast, the average similarities of quadrilaterals generally have no significant advantage over and may even be lower than the average level of explanation graphs. (2) Triangles are generally more important than quadrilaterals in both datasets. (3) The average similarity distribution of different triangle types are close. In contrast, interaction-related "cop" and "fi" based quadrilaterals are generally of higher similarities compared with social-related "fof" based quadrilaterals in both tower.

Table 6. NDCG@10-based fidelity scores across four datasets.

Method	Yelp	Flickr	Ciao	LastFM
GSAT	2.37	7.88	9.53	3.18
SoREX w/ top-K ego-path	2.75	8.13	12.42	4.01
SoREX w/ IB constraint	3.87	8.91	14.26	6.03
SoREX w/o re-aggr	3.95	8.74	15.29	5.81
SoREX	**5.77**	**10.38**	**20.83**	**7.27**

Quantitative Analysis. We adopt fidelity scores to quantitatively demonstrate the effectiveness of our explanations. Fidelity score measures the performance drop when important features are removed. In our situation, if the model can capture more important ego-paths for each user-item pair, the removal of such ego-paths would lead to more significant performance drop. Considering that GSAT [30] only uses sampled explanatory subgraph for prediction just like our SoREX, we compare SoREX with GSAT and define the fidelity score based on NDCG score. Given user-item pair (u_i, v_j), we first re-aggregate the ego-path subset $\hat{\mathcal{W}}^*_{ego} \backslash \tilde{\mathcal{W}}^*_{ego}(j)$ that are not sampled in each tower to obtain new predicted result $\hat{NDCG}_{ij}$. Then, we calculate the percentage of its performance drop compared with original NDCG@10 result $NDCG_{ij}$. Therefore, the fidelity score $\Delta NDCG\%$ is formulated as $\Delta NDCG\% = \frac{1}{|\mathcal{E}_t|} \sum_{(u_i,v_j)\in\mathcal{E}_t} \frac{(NDCG_{ij} - \hat{NDCG}_{ij})}{NDCG_{ij}}\%$. Similar procedures are also conducted for GSAT. The results on all four datasets are listed in Table 6. Compared with GSAT, SoREX can learn factor-specific and candidate-aware ego-path similarity distributions, while GSAT assigns the same weight for each node regardless of candidate items. Meanwhile, without the re-aggregation module (w/o re-aggr), the model's fidelity scores decrease significantly, highlighting the critical role of the re-aggregation module in aligning interpretability with prediction results. In addition, the top-K and IB constraint variants show further drops in fidelity, due to their hard truncation and information compression, respectively, both of which lead to loss of important

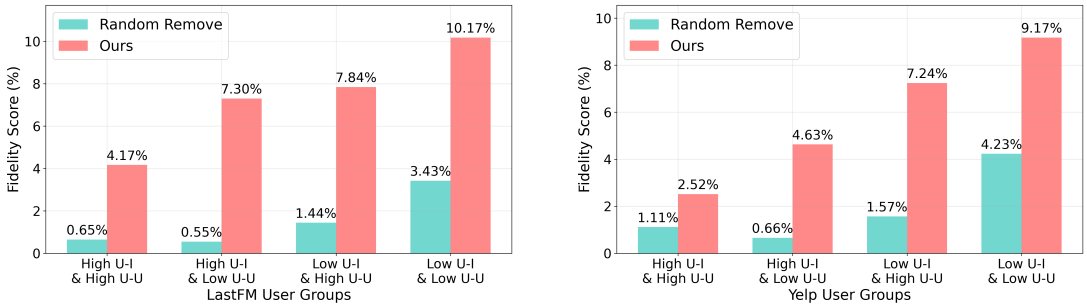


Fig. 10. Fidelity analysis across users with different characteristics. The fidelity of explanations is analyzed for user groups with different user–item and user–user degrees on Yelp and LastFM.

explanatory signals. Thus, our SoREX can provide more appropriate and personalized explanations and consistently achieve higher fidelity scores.

We further conduct a more in-depth analysis on the LastFM and Yelp datasets by examining the fidelity scores of users with different item graph and social graph degrees, in comparison with random path removal. Specifically, users are grouped into four categories based on whether their user–item (u–i) and user–user (u–u) degrees are above or below the median (i.e., 50th percentile), resulting in a 2×2 grouping: high u–i degree, low u–i degree, high u–u degree, and low u–u degree. As shown in Figure 10, for user groups with lower degrees, random removal of paths leads to a more significant drop in model performance. At the same time, our explanation paths consistently improve fidelity scores across all degree groups, with the benefit being most pronounced for users with low degrees. This suggests that for low-degree users, the identified ego-paths are particularly critical, as these users have fewer alternative paths. Consequently, providing faithful explanations for their recommendations is both more useful and more impactful.

6 Limitations and Future Directions

This work is an initial attempt to combine the concept of comparative explanation with graph-based social recommender explainer. Our proposed SoREX involves using relevant ego-path sampling to generate candidate-aware explanatory subgraphs. However, we believe there are still several challenges remaining:

Understandability. We propose to provide dense subgraph based explanations for detailed comparison among different candidate items. However, the graph-based explanations are not easy to understand, especially for those based on dense graphs. Although we could make the explanations more understandable by adopting certain post-processing programs or assigning real-world meanings to certain subgraph patterns, they are still not intuitive and precise enough. Balancing the precision and comprehensiveness of graph-based explanations remains an significant challenge.

Scalability. As we have analyzed in Section 4.6, we compromise the time and memory complexity of SoREX to achieve better explainability. However, when the social recommenders are deployed in industrial million-scale graphs, such trade-off would be unacceptable. How to develop comparatively explainable social recommenders with sub-linear complexity remains a challenge.

Evaluation of Explainer. Although multiple evaluation methods are adopted for the evaluation of SoREX, the lack of ground truth explanatory subgraphs makes it difficult to perform further quantitative verification. Such lack of ground

truth labels is not only the lack of labels themselves, but also the lack of proper label definitions. We could only explore some enlightening patterns based on certain subgraph templates, while the social recommendation explainers are highly likely to learn spurious correlations without the hints from external data. In fact, most social science related applications face similar limitations. Hence, we need an efficient and reliable strategy to evaluate self-explainable models without the access to ground truth explanations.

7 Conclusion

In this work, we introduce a novel self-explainable social recommendation framework SoREX to fill the explainability gap in GNN-based social recommendation. We first devise a social influence aware and friend recommendation enhanced two-tower framework to independently model the influence of social and interaction factors and lay the foundation for factor-specific explanation. Then, we propose to provide relevant ego-path based explanations. We transform ego-net of target user into a set of multi-hop ego-paths, and extract factor-specific and candidate-aware dense ego-path subsets for each candidate item, enabling comparative explanation among different candidates and factors via complex substructure investigation. We further perform explanation re-aggregation to emphasize relevant subgraphs of the user's ego-net and relate explanations to downstream predictions to make SoREX self-explainable. In addition, we design an auxiliary friend-recommendation task to capture more reliable friend relations and strengthen the social tower. Comprehensive experiments on four benchmark datasets demonstrate the effectiveness of SoREX in both predictive accuracy and explainability.

Acknowledgments

This research was supported by the Public Computing Cloud of Renmin University of China and by the Fund for Building World-Class Universities (Disciplines) at Renmin University of China.

References

[1] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-level Explanations. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. 1583–1592.

[2] Robert B Cialdini and Noah J Goldstein. 2004. Social influence: Compliance and conformity. *Annu. Rev. Psychol.* 55 (2004), 591–621.

[3] Enyan Dai and Suhang Wang. 2021. Towards Self-Explainable Graph Neural Network. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM '21)*. 302–311.

[4] Enyan Dai and Suhang Wang. 2024. Towards Prototype-Based Self-Explainable Graph Neural Network. *ACM Transactions on Knowledge Discovery from Data* (2024).

[5] Jeffrey Dastin. 2018. Amazon scraps secret AI recruiting tool that showed bias against women. *Reuters* (10 October 2018). https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G Accessed: 2025-07-01.

[6] Yuntao Du, Jianxun Lian, Jing Yao, Xiting Wang, Mingqi Wu, Lu Chen, Yunjun Gao, and Xing Xie. 2023. Towards Explainable Collaborative Filtering with Taste Clusters Learning. In *Proceedings of the ACM Web Conference 2023 (WWW '23)*. 3712–3722.

[7] Shaohua Fan, Xiao Wang, Yanhu Mo, Chuan Shi, and Jian Tang. 2024. Debiasing graph neural networks via learning disentangled causal substructure. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS '22)*. Article 1808, 13 pages.

[8] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *The World Wide Web Conference (WWW '19)*. 417–426.

[9] Matthias Fey and Jan Eric Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *Proceedings of the 7th International Conference on Learning Representations (RLGM Workshop) (ICLR '18)*.

[10] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. 2016. A Novel Recommendation Model Regularized with User Trust and Item Ratings. *IEEE Transactions on Knowledge and Data Engineering* 28, 7 (2016), 1607–1620.

[11] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 639–648.

[12] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 5–53.

[13] Alex Hern. 2017. YouTube and Google boycott spreads to US as AT&T and Verizon pull ads. *The Guardian* (23 March 2017). https://www.theguardian.com/technology/2017/mar/23/youtube-google-boycott-att-verizon-pull-adverts-extremism Accessed: 2025-07-01.

[14] Zheng Hu, Satoshi Nakagawa, Liang Luo, Yu Gu, and Fuji Ren. 2023. Celebrity-Aware Graph Contrastive Learning Framework for Social Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)*. 793–802.

[15] Rundong Huang, Farhad Shirani, and Dongsheng Luo. 2024. Factorized explainer for graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 38. 12626–12634.

[16] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*. 135–142.

[17] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*.

[18] Ke Ji and Hong Shen. 2016. Jointly modeling content, social network and ratings for explainable and cold-start recommendation. *Neurocomputing* 218 (2016), 1–12.

[19] Wei Jiang, Xinyi Gao, Guandong Xu, Tong Chen, and Hongzhi Yin. 2024. Challenging Low Homophily in Social Recommendation. In *Proceedings of the ACM Web Conference 2024 (WWW '24)*. 3476–3484.

[20] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[21] Jiuqiang Li and Hongjun Wang. 2024. Graph Diffusive Self-Supervised Learning for Social Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*. 2442–2446.

[22] Lei Li, Jianxun Lian, Xiao Zhou, and Xing Xie. 2024. Ada-retrieval: An adaptive multi-round retrieval paradigm for sequential recommendations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8670–8678.

[23] Lei Li, Xiao Zhou, and Zheng Liu. 2025. R2MED: A Benchmark for Reasoning-Driven Medical Retrieval. *arXiv preprint arXiv:2505.14558* (2025).

[24] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural Rating Regression with Abstractive Tips Generation for Recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. 345–354.

[25] Zongwei Li, Lianghao Xia, and Chao Huang. 2024. RecDiff: Diffusion Model for Social Recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24)*. 1346–1355.

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

[27] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized explainer for graph neural network. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS'20)*.

[28] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM '11)*. 287–296.

[29] Yijun Ma, Chaozhuo Li, and Xiao Zhou. 2024. Tail-steak: Improve friend recommendation for tail users via self-training enhanced knowledge distillation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 38. 8895–8903.

[30] Siqi Miao, Mia Liu, and Pan Li. 2022. Interpretable and generalizable graph learning via stochastic attention mechanism. In *International Conference on Machine Learning*. 15524–15543.

[31] Siqi Miao, Yunan Luo, Mia Liu, and Pan Li. 2022. Interpretable Geometric Deep Learning via Learnable Randomness Injection. In *The Eleventh International Conference on Learning Representations*.

[32] Peter Müller, Lukas Faber, Karolis Martinkus, and Roger Wattenhofer. [n. d.]. GraphChef: Decision-Tree Recipes to Explain Graph Neural Networks. In *The Twelfth International Conference on Learning Representations (ICLR '24)*.

[33] Aravind Sankar, Yozen Liu, Jun Yu, and Neil Shah. 2021. Graph Neural Networks for Friend Ranking in Large-scale Social Platforms. In *Proceedings of the Web Conference 2021 (WWW '21)*. 2535–2546.

[34] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[35] Sangwoo Seo, Sungwon Kim, and Chanyoung Park. 2023. Interpretable prototype-based graph information bottleneck. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*. 76737–76748.

[36] Caihua Shan, Yifei Shen, Yao Zhang, Xiang Li, and Dongsheng Li. 2021. Reinforcement Learning Enhanced Explainer for Graph Neural Networks. In *Proceedings of the 35th International Conference on Neural Information Processing Systems (NeurIPS'21, Vol. 34)*. 22523–22533.

[37] Xiran Song, Jianxun Lian, Hong Huang, Mingqi Wu, Hai Jin, and Xing Xie. 2022. Friend Recommendations with Self-Rescaling Graph Neural Networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. 3909–3919.

[38] Yongduo Sui, Xiang Wang, Jiancan Wu, Min Lin, Xiangnan He, and Tat-Seng Chua. 2022. Causal Attention for Interpretable and Generalizable Graph Classification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. 1696–1705.

[39] Yiyi Tao, Yiling Jia, Nan Wang, and Hongning Wang. 2019. The FacT: Taming Latent Factor Models for Explainability with Factorization Trees. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. 295–304.

[40] Ye Tao, Ying Li, Su Zhang, Zhirong Hou, and Zhonghai Wu. 2022. Revisiting Graph Based Social Recommendation: A Distillation Enhanced Social Graph Network. In *Proceedings of the ACM Web Conference 2022 (WWW '22)*. 2830–2838.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30.

[42] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable Recommendation via Multi-Task Learning in Opinionated Text Data. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. 165–174.

[43] Xiting Wang, Yiru Chen, Jie Yang, Le Wu, Zhengtao Wu, and Xing Xie. 2018. A Reinforcement Learning Framework for Explainable Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. 587–596.

[44] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5329–5336.

[45] Zhen Wang, Bo Zong, and Huan Sun. 2021. Modeling Context Pair Interaction for Pairwise Tasks on Graphs. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM '21)*. 851–859.

[46] Jiahao Wu, Wenqi Fan, Jingfan Chen, Shengcai Liu, Qing Li, and Ke Tang. 2022. Disentangled Contrastive Learning for Social Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM '22)*. 4570–4574.

[47] Le Wu, Junwei Li, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2020. Diffnet++: A neural influence and interest diffusion network for social recommendation. *IEEE Transactions on Knowledge and Data Engineering* 34, 10 (2020), 4753–4766.

[48] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A Neural Influence Diffusion Model for Social Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. 235–244.

[49] Le Wu, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2021. Collaborative Neural Social Recommendation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51, 1 (2021), 464–476.

[50] Yikun Xian, Zuohui Fu, S. Muthukrishnan, Gerard de Melo, and Yongfeng Zhang. 2019. Reinforcement Knowledge Graph Reasoning for Explainable Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. 285–294.

[51] Aobo Yang, Nan Wang, Renqin Cai, Hongbo Deng, and Hongning Wang. 2022. Comparative Explanations of Recommendations. In *Proceedings of the ACM Web Conference 2022 (WWW '22)*. 3113–3123.

[52] Aobo Yang, Nan Wang, Hongbo Deng, and Hongning Wang. 2021. Explanation as a Defense of Recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM '21)*. 1029–1037.

[53] Bo Yang, Yu Lei, Jiming Liu, and Wenjie Li. 2016. Social collaborative filtering by trust. *IEEE transactions on pattern analysis and machine intelligence* 39, 8 (2016), 1633–1647.

[54] Yonghui Yang, Le Wu, Zihan Wang, Zhuangzhuang He, Richang Hong, and Meng Wang. 2024. Graph Bottlenecked Social Recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*. 3853–3862.

[55] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. 2019. On the (In)fidelity and Sensitivity of Explanations. In *Advances in Neural Information Processing Systems*, Vol. 32.

[56] Haoteng Yin, Muhan Zhang, Yanbang Wang, Jianguo Wang, and Pan Li. 2022. Algorithm and System Co-Design for Efficient Subgraph-Based Graph Representation Learning. *Proc. VLDB Endow.* 15, 11 (2022), 2788–2796.

[57] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. GNNExplainer: Generating Explanations for Graph Neural Networks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS'19, Vol. 32)*.

[58] Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. 2020. Graph Information Bottleneck for Subgraph Recognition. In *International Conference on Learning Representations*.

[59] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. 2021. Socially-Aware Self-Supervised Tri-Training for Recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21)*. 2084–2092.

[60] Junliang Yu, Hongzhi Yin, Jundong Li, Min Gao, Zi Huang, and Lizhen Cui. 2022. Enhancing Social Recommendation With Adversarial Graph Convolutional Networks. *IEEE Transactions on Knowledge and Data Engineering* 34, 8 (2022), 3727–3739.

[61] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021. Self-Supervised Multi-Channel Hypergraph Convolutional Network for Social Recommendation. In *Proceedings of the Web Conference 2021 (WWW '21)*. 413–424.

[62] Xiao Yu, Xiang Ren, Yizhou Sun, Bradley Sturt, Urvashi Khandelwal, Quanquan Gu, Brandon Norick, and Jiawei Han. 2013. Recommendation in heterogeneous information networks with implicit user feedback. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. 347–350.

[63] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. 2020. XGNN: Towards Model-Level Explanations of Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. 430–438.

[64] Jiaxing Zhang, Dongsheng Luo, and Hua Wei. 2023. MixupExplainer: Generalizing Explanations for Graph Neural Networks with Data Augmentation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*. 3286–3296.

[65] Yongfeng Zhang, Xu Chen, et al. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval* 14, 1 (2020), 1–101.

[66] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 83–92.

[67] Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Cheekong Lee. 2022. Protgnn: Towards self-explaining graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 9127–9135.

[68] Kangzhi Zhao, Xiting Wang, Yuren Zhang, Li Zhao, Zheng Liu, Chunxiao Xing, and Xing Xie. 2020. Leveraging Demonstrations for Reinforcement Recommendation Reasoning over Knowledge Graphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 239–248.

[69] Tianxiang Zhao, Dongsheng Luo, Xiang Zhang, and Suhang Wang. 2023. Towards Faithful and Consistent Explanations for Graph Neural Networks. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23)*. 634–642.

[70] Xiao Zhou, Cecilia Mascolo, and Zhongxiang Zhao. 2019. Topic-enhanced memory networks for personalised point-of-interest recommendation. In *Proceedings of the 25th ACM SIGKDD International conference on knowledge discovery & data mining*. 3018–3028.

[71] Xiao Zhou, Zhongxiang Zhao, and Hanze Guo. 2025. Tricolore: Multi-Behavior User Profiling for Enhanced Candidate Generation in Recommender Systems. *IEEE Transactions on Knowledge and Data Engineering* (2025).

[72] Huaisheng Zhu, Dongsheng Luo, Xianfeng Tang, Junjie Xu, Hui Liu, and Suhang Wang. 2023. Self-Explainable Graph Neural Networks for Link Prediction. *arXiv preprint arXiv:2305.12578* (2023).