Knowledge-Based Systems 251 (2022) 109187

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

GraphRR: A multiplex Graph based Reciprocal friend Recommender system with applications on online gaming service

Yaomin Chang [a,b], Lin Shu [a,b], Erxin Du [a,b], Chuan Chen [a,b,*], Ziyang Zhang [a,b], Zibin Zheng [a,b], Yuzhao Huang [c], Xingxing Xing [c]

[a] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[b] National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou, China
[c] UX Center, NetEase Games, Guangzhou, China

ARTICLE INFO

Article history:
Received 9 February 2022
Received in revised form 9 May 2022
Accepted 30 May 2022
Available online 15 June 2022

Keywords:
Reciprocal Recommender System
Graph Neural Network
Multiplex graph

ABSTRACT

Reciprocal Recommender Systems (RRSs) are recommender systems specifically designed for people-to-people recommendation tasks, e.g., online gaming, dating, and recruitment services. They are fundamentally different from the conventional user–item recommendations. In RRSs, user interactions are usually directional, i.e., they are initiated by one side and not necessarily reciprocated by the other side. In the meanwhile, abundant multiplex user interactions, e.g., *Friend Request* and *Send Message*, are collated by the online services and can be represented into a large-scale multiplex user interaction graph. Despite the substantial progress of Graph Neural Networks (GNNs) on capturing users' multiplex interactions, naive GNNs are insufficient to capture the additional information implied from the directions of interactions, as they are usually not designed to preserve the asymmetric proximities between users.

In the paper, we present a novel Graph neural network for Reciprocal Recommendation (GraphRR) to utilize the multiplex user interactions. Specifically, three ego graphs are augmented based on the directions of interactions for each user to capture his preference, attraction and similarity in a finer granularity. Then the multiplexity-aware GNN modules are further applied to measure the contributions of different interaction types. Extensive experiments are conducted in the datasets of the real-world large-scale online games from *NetEase Games*, a leading game provider for worldwide users. The experimental results demonstrate the superiority of GraphRR over baseline methods and provide empirical evidence for the benefits of the proposed ego graph augmentation. The source code is also available online for reproductivity[1].

1. Introduction

Recommender Systems (RSs) have been widely developed and employed in a variety of applications including E-commerce shopping systems [1,2], social media [3], and online video service [4], etc. However, these conventional user–item RSs are incapable of accommodating people-to-people recommendation scenarios, as they only consider users' unidirectional preference toward inanimate items. While the recommended items cannot "reject" users in the classical RSs, unilateral behaviors occur frequently in the people-to-people recommendation.

Reciprocal Recommender Systems (RRSs) aim to connect users with mutual preference, i.e., reciprocity, in the people-to-people recommendation scenarios. For instance, there are some RRSs in online games designed for recommending mutually preferable teammates/friends to increase gamer's engagement. Fig. 1 shows an example of RRS on the online game *Knives Out*, which is a popular mobile game in China. When gamers require the recommendation result, the recommender system aims to provide a user list that is appealing to users from both sides. Apart from the online gaming service, there also exist many other important domains that involve reciprocal recommendations, including online social platform [5], online dating [6] and recruitments [7], etc. In spite of the urgent requirements of these applications, RRSs have received little attention compared to the conventional recommender system.

* Corresponding author at: School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China.
E-mail addresses: changym3@mail2.sysu.edu.cn (Y. Chang), shulin@mail2.sysu.edu.cn (L. Shu), duerx@mail2.sysu.edu.cn (E. Du), chenchuan@mail.sysu.edu.cn (C. Chen), zhangzy233@mail2.sysu.edu.cn (Z. Zhang), zhzibin@mail.sysu.edu.cn (Z. Zheng), huangyuzhao@corp.netease.com (Y. Huang), xingxingxing@corp.netease.com (X. Xing).

1 https://github.com/changym3/GraphRR

https://doi.org/10.1016/j.knosys.2022.109187

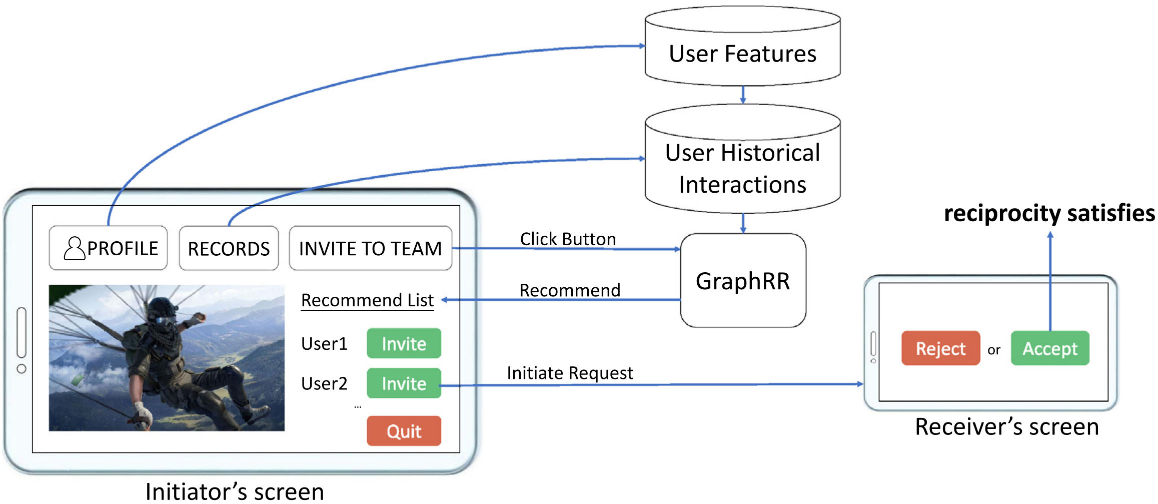


Fig. 1. An example of the RRS in the online game service *Knives Out*. The left screen is the home page of the gamers, where they may click the "INVITE TO TEAM" button to find teammates from the system and then initiate an invitation request to other users. The right screen shows the receiver's response to the request. When the first gamer clicks the "Invite" button and the second gamer clicks the "Accept" button, the reciprocity is satisfied in this recommendation.

The objective of RRSs is to create a mutually preferable match between two users, which renders the task of reciprocal recommendation more complex and challenging. The RRS research in the literature can be generally divided into two categories: Content-Based (CB) and Collaborative Filtering (CF). The CB approaches [6] are primarily based on the description of items and user profiles, thus heavily relying on the attributes by feature engineering. The CF-based approaches [8,9] propose to capture the collaborative signals between similar users based on their historical interactions. The limitation of CF-based methods lies in their non-scalability due to the extravagant calculation of the similarities over all pairs of users. A recent work [10] takes advantage of the latent factor model to generate user embeddings from the dual interaction matrix, showing significant improvement on both performance and computational efficiency compared to other CF-based approaches. Nevertheless, it is still limited in dating scenarios, which inherently have two clear roles (male and female) to be matched. It cannot generalize well to some complex reciprocal scenarios, e.g., the online gaming service.

As user interactions can be intrinsically described in graph-structured data, where nodes represent users/items and edges represent users' interactive behaviors, there have been many prominent studies [3,11,12] incorporating graph-based techniques into recommender systems. These methods leverage Graph Neural Networks (GNNs) [13], an emerging paradigm for graph representation learning, to effectively generate user representations. One of the significant advantages is their expressive capacities to jointly learn node attributes and graph structures. Another advantage lies in their strong extensibility to tackle the multiplexity of graphs, where multiple types of edges may exist between two nodes. Recently, some works [14,15] have also demonstrated the effectiveness of multiplex GNNs in the user interaction graph, facilitating the recommender systems to explore more fine-grained user portraits.

Despite their promising success in conventional recommender systems, GNNs remain relatively unexplored in the large-scale reciprocal recommender system. It is beneficial to combine GNNs' capacity into RRS to determine users' reciprocity. However, designing GNNs for reciprocal recommender systems poses unique challenges. First, the requirement of reciprocity cannot be directly fulfilled by applying GNNs on the directional interaction graph as in the conventional user–item recommendation scenarios. We will show a detailed analysis in the paper that user interaction graphs declare the properties of heterophily, asymmetry, and non-transitivity. These properties render traditional GNNs insufficient to capture the information implied from the directions of interactions. Second, users continually generate multiplex interactions, which reveal different aspects of user portraits in the online service. The measurement of contributions of multiplex interaction needs to be considered by the RRS.

Concerning the challenges mentioned above, in this work, we propose a carefully designed multiplex Graph based Reciprocal friend Recommender system (GraphRR) with applications on two online games to enhance gamers' engagement. Here we propose five research questions to guide readers to introduce our works and contributions:

- **RQ_1** Why common recommendation methods are insufficient to capture users' portraits?
- **RQ_2** Why multiplexity is important in the scenarios of recommendation?
- **RQ_3** Why does the direction of each interaction need to be considered in the reciprocal recommendation?
- **RQ_4** Why GNNs cannot be directly applied in the directional user interaction graph for the reciprocal recommendation?
- **RQ_5** How to utilize GNNs in the task of reciprocal recommendation?

To summarize, our key contributions are three-fold:

- In view of the unidirectionality of the user interaction graph, we analyze its property of heterophily, asymmetry, and non-transitivity, which is handled by the proposed mechanism of the ego graph augmentation.
- We provide a GNN-based solution for RRSs, namely GraphRR, to capture users' preference, attraction and similarity separately. Besides, the attentive aggregation is integrated into GraphRR to further utilize the multiplex users' interactions in the online service.
- We conduct extensive experiments on the datasets from real-world online games and demonstrate that the proposed GraphRR outperforms a series of baseline methods.

The remainder of the paper is structured as follows: Section 2 discusses the related work. Section 3 formulates the task of the reciprocal recommendation and describes the notations used in the paper. In Section 4, we provide the analyses that motivate the designs of the proposed GraphRR. In Section 5, we introduce the details of the proposed GraphRR, including the ego-graph augmentation and the multiplex graph neural networks. Experimental results are shown in Section 6 to verify the effectiveness of the proposed method. Finally, the conclusion and future work are discussed in Section 7.

2. Related work

In this section, we review studies related to the proposed method, including the convectional user–item recommendation, the reciprocal recommendation, and the recently emerging graph-based recommendation.

2.1. User–item recommendation

The purpose of the user–item recommendation is to proactively provide customized items for users by extrapolating their preferences towards items. The user–item recommender systems are mainly categorized into collaborative filtering, content-based and hybrid approaches [16]. The content-based recommendation is primarily based on the auxiliary characteristics of users' and items' content, including texts, images, and videos. Collaborative filtering is one of the most widely used and successful techniques in modern recommender systems, which analyzes the similar interests among users' historical interactions [17,18]. Hybrid approaches are the combinations of content-based and collaborative filtering techniques with different strategies. With the development of deep neural networks, NeuMF [19] seamlessly combines the matrix factorization and the multi-layer perceptron into the recommender system. NAIS [20] introduces the attention mechanism to differentiate the contributions of items.

Besides, the multi-behavior recommender system is an emerging branch in the research of the user–item recommendation, where there are multiple types of user behaviors or interactions to be leveraged. Some previous works usually extend conventional recommender systems into multiple user behaviors to capture different semantics [21–23]. Specifically, CMF [21] proposes to decompose the user interaction matrices of different behaviors simultaneously. MF-BPR [22] adapts the sampling strategy with interaction-aware sampling probability to handle multiple user feedbacks. NMTR [23] solves the problem of multi-behavior recommendation under the paradigm of multi-task learning. Recently, MBGCN [24] proposes to represent multiple user behaviors into a unified multiplex graph and leverages graph convolutional networks to obtain user embeddings by applying the graph aggregation discriminately.

Though these methods have been demonstrated effective in the user–item recommendation scenarios, but they cannot be directly applied in the reciprocal setting as they only consider the unidirectional preference of users towards items.

2.2. Reciprocal recommendation

There are various approaches of RRSs being investigated, which can be broadly divided into two categories, content-based (CB) and collaborative filtering based (CF). One of the best-known studies of CB methods for RRS is RECON [6]. RECON is designed for the dating scenario, and firstly calculates users' unidirectional preference scores to each other and combines these two scores into a reciprocity score by harmonic mean. As CF-based methods achieve promising results in the user–item recommendation, some studies [5,8,9] also introduce the collaborative filtering technique into RRS solutions. However, it cannot scale into real-world datasets that contain millions of users [25]. Recently, LFRR [10] proposes to extrapolate users' latent vectors based on matrix factorization. Concretely, LFRR constructs two preference matrices: *female-to-male preferences* and *male-to-female preferences*. LFRR trains two latent factor models and estimates the reciprocity score by the dot product of latent vectors, which greatly reduces the computational complexity of collaborative filtering. However, current RRS methods still have simple model architectures and limited expressive capacities. Moreover, they only consider the first-order interactions and cannot capture the high-order user semantics.

2.3. Graph-based recommendation

Motivated by the recent success of Graph Neural Networks (GNNs), much effort has been devoted to applying GNNs on recommendations [3,11,26]. For example, GCMC [26] takes advantage of the graph auto-encoder, which firstly encodes users and items by the differentiable message propagation on the user–item graph and decodes the latent representations to complete the rating matrix in recommendations. PinSage [3] integrates both graph structures and node attributes by combining random walks and graph convolutional networks [27] to facilitate recommendations on web-scale graphs. NGCF [11] explores the high-order connectivities in the user–item bipartite graph by performing embedding propagation. LightGCN [28] simplifies GCN for the recommendation task by removing the transformation and nonlinear activation. Besides, there also emerges some studies [29] recently on the representation learning of heterogeneous graphs, which can be applied on the multiplex graph for the recommendation task. Despite the successful efforts on graph representation learning, there are still no previous studies that incorporate the powerful multiplex graph neural networks into RRSs to estimate the reciprocity score between users.

3. Problem formalization and notations

In this section, we first formulate the user interactions with multiple types as a multiplex graph, and specify the definition of reciprocal friends recommendation problems in the application of online games. The notations used in the paper are summarized in Table 1.

3.1. Multiplex interaction graph

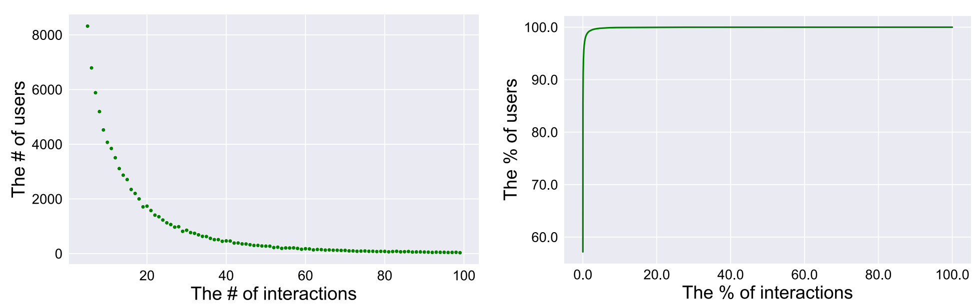
In real-world scenarios of the online social system, users can interact with other users in multiple manners, e.g., *Friend Request*, *Game Like*, *Send Gift*, etc. These user interactions are represented in a large-scale multiplex graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where $\mathcal{V}$ denotes the user set and $\mathcal{E}$ denotes the edge set. Each edge in $\mathcal{E}$ represents a directional interaction between users. As each user may have various types of interactions, the edge set $\mathcal{E}$ can be specialized as $\mathcal{E} = \mathcal{E}_{t_1} \cup \mathcal{E}_{t_2} \cup \cdots \cup \mathcal{E}_{t_T}$, where $\mathcal{E}_{t_k}$ denotes the set of users' interactions under the t_k type . The set of all interaction types are denoted as $\mathcal{T}_E = \{t_1, t_2, \ldots, t_T\}$ and there are T types of user interactions totally. Besides, each user is associated with an attribute representation, jointly denoted in the matrix form as $\mathbf{X} \in \mathbb{R}^{N \times F}$, where N is the number of users and F is the dimension of attribute representation. The user features usually reveal some significant patterns of their engagement in the platform, e.g., the user's competitive performance and social activity in the online gaming scenario.

3.2. Reciprocal friend recommendation

The reciprocal friend recommendation is to recommend mutually preferable users to each other for co-participating in some social activities, e.g., co-gaming or friending in our application. The reciprocal friend recommendation problem can be formulated into a link prediction task in the multiplex graph. Given user

Table 1
The notations in the paper.

Notations	Explanations
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$	Graph
$\mathcal{V}$	The user set in the graph
$\mathcal{E}$	The multiplex interaction set in the graph
$\mathbf{X}$	The user attribute matrix
t	Interaction type
$\mathbf{W}^{(l)}$	The transformation matrix in the lth layer
H, h	The user hidden representation in the form of matrix or vector
$\mathcal{E}_t$	Interaction set under interaction type t in the graph
$\mathcal{N}_{u,t}$	The u's neighbors under interaction type t
$\mathcal{E}_{u,t}$	The u's interaction set under interaction t
(src, dst)	Interaction between users src and dst
$\mathcal{E}^{seed}_{u,t}$	The t-type seed interactions of user u in reciprocity augmentation
$\mathcal{E}^{knn}_{u,t}$	The t-type kNN co-interactions of user u in reciprocity augmentation
$s_{(src,dst),t}$	The unnormalized importance coefficient of the t-type interaction (src, dst)
$\alpha_{(src,dst),t}$	The normalized importance coefficient of the t-type interaction (src, dst)
$e^*_{u,sim}$	The user u's representation in the similarity graph.
$e^*_{u,pref}$	The user u's representation in the preference graph.
$e^*_{u,attr}$	The user u's representation in the attraction graph.
e^*_{u}	The user u's ultimate representation.



(a) Distribution (b) Cumulative Distribution

Fig. 2. The distribution and the cumulative distribution of the interaction *Approve Friend Request*.

pairs (u, v) and type of the target interaction $t \in \mathcal{T}_E$, the reciprocal recommender should exploit the users' preferences in advance by utilizing all kinds of user historical interactions $\{t_1, t_2, \dots, t_T\}$ and recommend the most mutually preferable candidate v to u. Formally, the label of link prediction can be defined as whether these two users have the target interaction of t in a period, which be expressed as

$$y^t_{uv} = \begin{cases} 1, & \text{if } u \text{ has the interaction with } v \text{ under behavior } t; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Given the historical interactions of all users as well as the label of user pairs (u, v), the recommendation model aims to estimate the probability that user u and v interact with each other under the target positive interaction t after the system matching in the co-participation of some activities.

4. Analyses

In this section, we provide the data exploratory analysis to demonstrate the importance of considering multiplex interactions and interaction directions. Besides, we also provide a theoretical analysis of GNNs for modeling reciprocity.

4.1. The data analysis

Before we formalize our approach, we conduct an exploratory data analysis of the user interaction graph in our scenario, the online game service. The exploratory data is collected from the training data of *Knives Out*. The detailed data statistics are described in Section 6.

Long-tail distribution of interaction (**RQ$_1$**). Fig. 2(a) shows the distribution and cumulative distribution of the interaction *Approve Friend Request*. It is observed that these quantities approximately follow the power law, which indicates that a small portion of users contributes a large number of interactions. If we only leverage *Approve Friend Request* to predict whether users approve friend requests from others, this skewed distribution and the sparsity of the training data and labels leads to the degradation of the model's predictive abilities for the long tail users. To solve the problem, we could utilize the auxiliary interaction such as *Accept Team Invitation*, *Game Like*, which usually are more frequent in the game, to assist the training of sparse target labels.

The diverse homophily distribution over multiplexity (**RQ$_2$**). This empirical study is to validate whether the multiplexity information is distinctive. We begin by characterizing *multiplex homophily* on each normalized feature with respect to the topology of different interaction types. We define $s^t_{(src,dst)}$ as the interaction homophily between a user src and another user dst, where src has an t-type interaction with dst. Then the multiplex homophily on the features for interaction t is computed by the mean over all interactions of this type, define by:

$$s_t = \frac{1}{|\mathcal{E}_t|} \sum_{(src,dst)\in\mathcal{E}_t} s^t_{(src,dst)} \qquad s^t_{(src,dst)} = |\mathbf{x}_{src} - \mathbf{x}_{dst}|. \tag{2}$$

For simplicity, we only visualize the multiplex homophily on the first ten features for all interactions in Fig. 3. It is observed that the homophily varies significantly over multiplexity for a

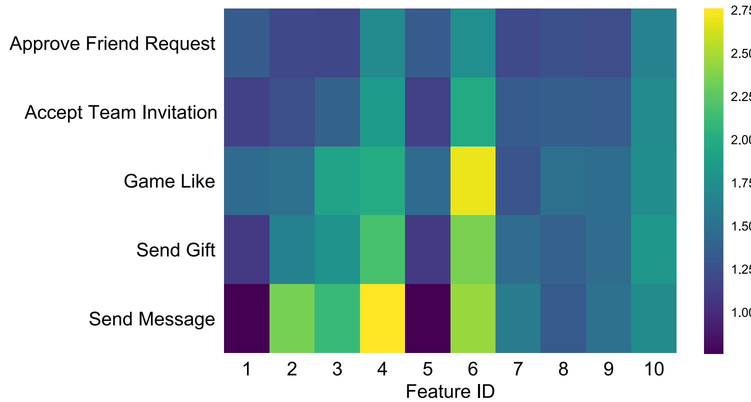


Fig. 3. The diverse homophily distribution implied from the multiplex topology of interactions.

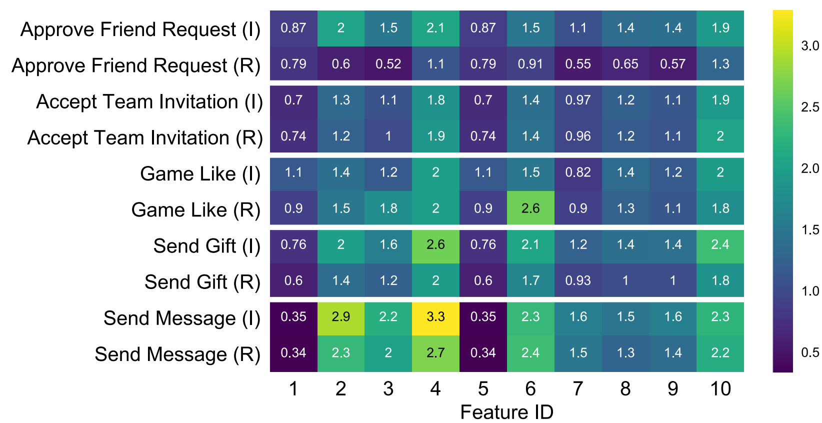


Fig. 4. The diverse homophily distribution implied from the directionality of interactions. The letter "I" denotes the interaction initiator and "R" denotes the interaction receiver.

majority of features, indicating the diverse distribution of user patterns with respect to the topological homophily.

The diverse homophily distribution over directionality (**RQ**$_3$). This empirical study is to validate whether the directionality reveals various semantics. We investigate the average features of the interaction initiator and receiver and show the statistics in Fig. 4. Evidently, substantial feature difference between the initiators and receivers is observed to a varied extent regarding the interaction type. The discrimination of initiators and receivers is informative as they reveal distinctive semantics of personalized social patterns. For example, a gamer who receives quantities of *Friend Request*s from other gamers probably owns superb competitive skills in the online gaming service. This analysis verifies the various engagement pattern of the interaction initiators and receivers, which motivate our model design in the reciprocal setting.

4.2. The analysis of GNNs for reciprocity (**RQ**$_4$)

As stated before, many user interactions are directional, implying that only a single side shows willingness while the other side may be indifferent even repulsive to his enthusiasm.

The directionality of user interactions has a twofold effect. On the one hand, it provides a finer granularity to describe user profiles from two dimensions, i.e., preference and attraction. As shown in Fig. 5, user u_1 shows his attraction to $\{u_2, u_3, u_4\}$ while he only prefers $\{u_5, u_6, u_7, u_8\}$. The preference and attraction of each user can be inferred from the interaction initiators and receivers, and they reveal distinctive user portraits. On the other hand, the directionality and the intrinsic semantics of user

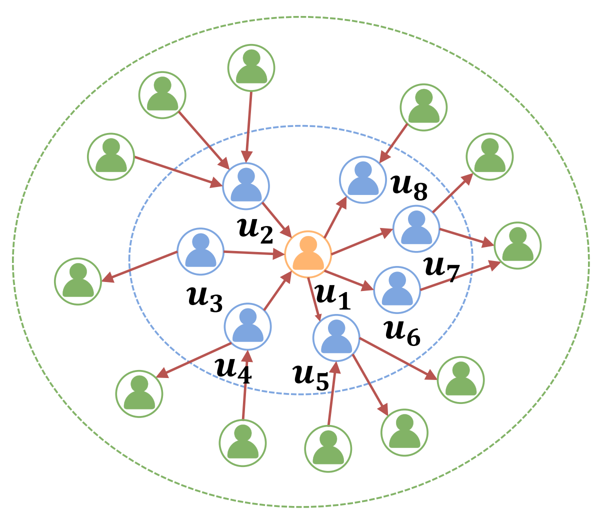


Fig. 5. A toy example of the directional user interaction graph for a single user.

interactions result in the properties of **heterophily** [30], **asymmetry** [31], and **non-transitivity** [32] of user interaction graphs. The property of heterophily declares that the connected nodes in the graph are likely from different classes or have dissimilar features. For nodes a, b and c in a graph, the property of asymmetry declares that the proximity between (a, b) is not equal to that of (b, a), and the property of non-transitivity declares that $a \rightarrow c$ may not hold when $a \rightarrow b$ and $b \rightarrow c$ are satisfied.

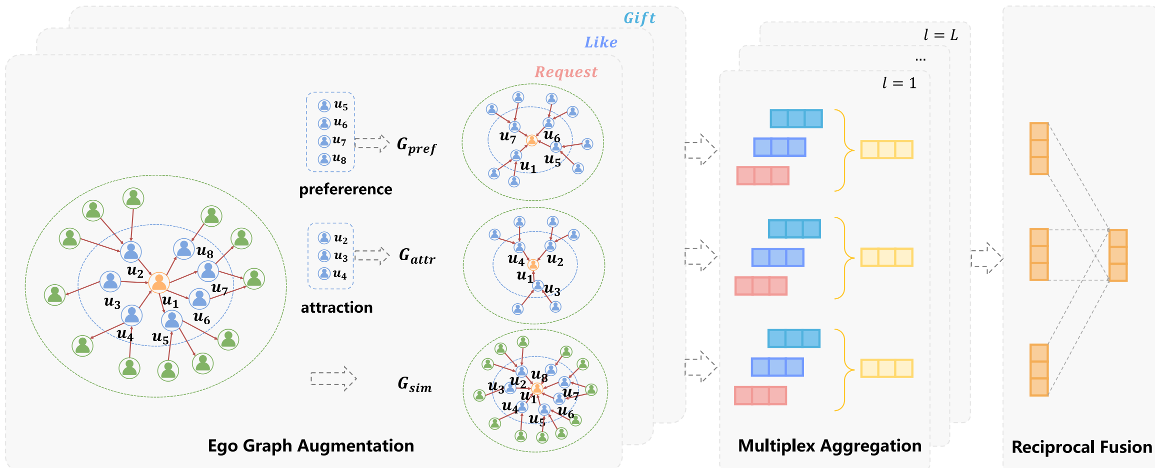


Fig. 6. The overall architecture of GraphRR. The ego graph augmentation is illustrated in the left part of the figure, which generates the preference, attraction and similarity ego graphs for each user. In each layer of the ego graph, the interaction-specific embeddings are generated and attentively combined. GraphRR fuses the representations learned from three augmented graphs into the ultimate embedding by the reciprocal information fusion.

In our scenario of RRSs, it is also intuitive that gamers often prefer teammates with others whose competitive styles are complementary to themselves (heterophily), their interactions could not be always reciprocated (asymmetry), and their directional preferences cannot transit from users to users (non-transitivity).

According to these analyses, we argue that the performance of GNNs for reciprocal tasks will be limited if they are directly employed on the user interaction graph without specific designs. First, some valuable fine-grained information of user attraction and preference implied from the edge directions are discarded and mixed up if they are regarded as bi-directed edges by GNNs. Second, GNNs play the role of low-pass filtering and smooth the node features over graph topological structures, where the commonality of node features are mainly retained and their difference are ignored [33,34]. Hence GNNs are effective when edges encode similarity but fail to explore the user interaction graph when edges encode heterophily and asymmetry. Third, the high-order information is empirically considered with multi-layer GNN architectures to achieve the best performance [27,35]. However, the high-order information may be noisy and entangled when considering the directions of interactions in the user interaction graph due to its non-transitivity. To summarize, the properties of heterophily, asymmetry, and non-transitivity of the user interaction graph render the task of reciprocal recommendation with GNNs challenging.

5. Methodology (RQ_5)

In this section, we introduce the details of the proposed GraphRR, which is illustrated in Fig. 6.

5.1. The ego graph augmentation

In view of the limitations of traditional GNNs for reciprocal tasks, we propose the mechanism of the ego graph augmentation. For each user, three L-hop ego graphs are augmented for L-layer GNNs, which encode the user preference, attraction and similarity respectively.

5.1.1. Reciprocity-based ego graphs

The reciprocity-based augmentation generates two ego graphs for each user, i.e. the preference ego graph and the attraction ego graph. Since directional user interaction graphs cannot be simply applied in GNNs due to the properties of asymmetry, heterophily and non-transitivity, we consider how to convert the directional interactions into symmetric, homophily-encoding and transitive ones while preserving their useful semantics. As shown in the left graph on Fig. 7, u_1 send *Friend Request*s to u_2 and u_3, we could find that u_2 and u_3 indeed share a new personalized relation named *Co-Requested by* u_1. Moreover, given an ego node u_1, this new relation satisfies the property of symmetry, homophily and transitivity. Hence we integrate this new relation into the constructed ego graph.

Each constructed ego graph consists of two components: the seed interactions and the kNN co-interactions. The former specifies the first-order neighbors and the latter specifies the higher-order neighbors to explore the kNN co-interaction semantics between users. An example of the reciprocity-based augmentation is shown in Fig. 7. For simplicity, we only introduce the construction of the preference ego graph, and the attraction ego graph can be obtained in a similar way. We now elaborate on the construction of the two components of the preference ego graph as below.

The seed interactions. The seed interactions of user u's preference ego graph is defined as his direct out-going interactions. Formally, given user u and the interaction type t, we firstly obtain his first-order neighbors as

$$\mathcal{N}_{u,t}^{seed} = \{v || u \xrightarrow{t} v\}. \tag{3}$$

Then the seed interactions are constructed as

$$\mathcal{E}_{u,t}^{seed} = \{(u, v) || v \in \mathcal{N}_{u,t}^{seed}\}. \tag{4}$$

The kNN co-interactions. The kNN co-interactions are generated to explore the co-interaction semantics between users, which act as the higher-order neighbors of the constructed ego graphs. Given the seed nodes $\mathcal{N}_{u,t}^{seed}$, we firstly apply the k-nearest neighbor (kNN) algorithm within the node set to generate the kNN graph as

$$\mathcal{E}_{u,t}^{knn} = k\text{-nearest}(\{(src, dst) || src \in \mathcal{N}_{u,t}^{seed}, dst \in \mathcal{N}_{u,t}^{seed}\}), \tag{5}$$

where src is one of the k-nearest neighbors of dst, and the kNN distance between users is also recorded as edge features in $\mathcal{E}_{u,t}^{knn}$. Then the co-interaction edges are iteratively expanded from neighbors of the previous order based on the kNN connectivity:

$$\mathcal{E}_{u,t}^{(l)} = \{(src, dst) || (src, dst) \in \mathcal{E}_{u,t}^{knn}, dst \in \mathcal{N}_{u,t}^{(l-1)}\}, l \geq 2, \tag{6}$$

$$\mathcal{N}_{u,t}^{(l)} = \{src || (src, dst) \in \mathcal{E}_{u,t}^{(l)}\}, l \geq 2. \tag{7}$$

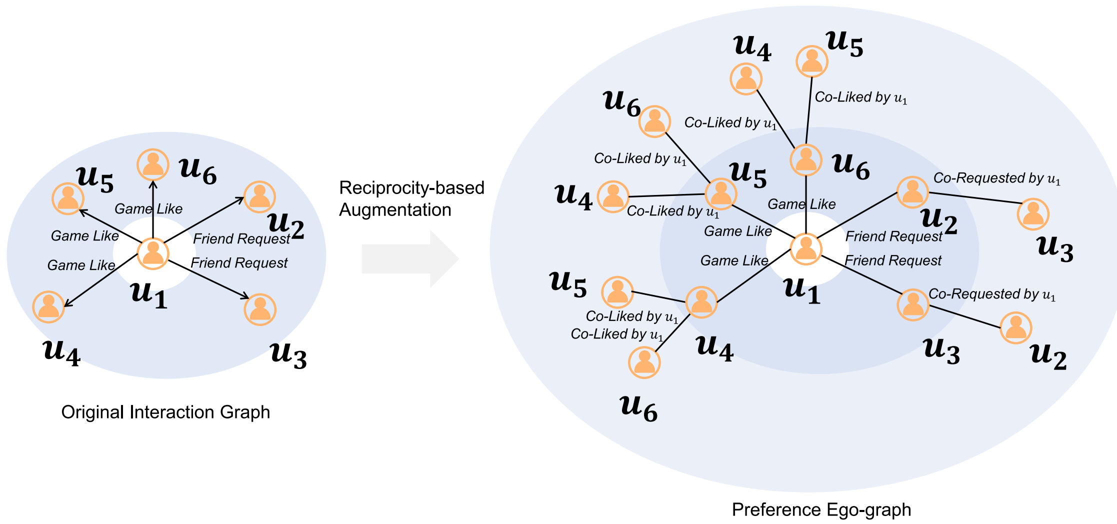


Fig. 7. An example of the reciprocity-based augmentation for the 2-hop preference ego graph. The left part is the original user interaction graph where only the out-going edges are drawn for simplicity. The right part is the augmented preference ego graph, which consists of the seed interactions (the dark circle) and the kNN co-interactions (the light circle).

For the case of $l = 1$,

$$\mathcal{E}_{u,t}^{(1)} = \mathcal{E}_{u,t}^{seed}, \tag{8}$$

$$\mathcal{N}_{u,t}^{(1)} = \mathcal{N}_{u,t}^{seed}. \tag{9}$$

There are two reasons to explain why we use kNN to construct the co-interactions. First, a user may have many interactions with other users, resulting in exponentially increasing neighborhood expansion and computational complexity if we do not limit the neighbor size. Second, though sampling techniques can be a compromised solution to the computation burden, it ignores the significant information contained in the user features during sampling, which evidently reveals users' engagement in the game, e.g., users' frequency of being online. The kNN algorithm is important to achieve the balance between effectiveness and efficiency. In terms of the distance metrics, there are multiple choices including Euclidean distance, cosine similarity, and heat kernel distance. Here the Euclidean distance is chosen since the user's attributes usually possess explicit meanings and we are more concerned about their numerical values rather than their vectorial direction.

5.1.2. Similarity-based ego graphs

Since GNNs could act as the role of low-pass filtering, we still leverage GNNs to extract the common components of users based on the topological structures of user interaction graphs. Hence, we construct the similarity-based ego graphs by neglecting the directions of interactions and expanding the L-hop neighborhood for each ego user. The aggregation on similarity ego graphs can also be viewed as directly applying GNNs to the original user interaction graph.

5.2. Multi-interaction GNN

In the proposed GraphRR, three augmented ego graphs for user u are generated denoted as $\mathcal{G}_{pref}(u)$, $\mathcal{G}_{attr}(u)$ and $\mathcal{G}_{sim}(u)$, where three independent Multi-Interaction GNN (MIGNN) modules are applied to obtain user representations in the aspect of preference, attraction and similarity, respectively. Here we describe the MIGNN by introducing the details of message passing in the augmented ego graphs. A single MIGNN-layer contains two types of aggregation, i.e., the user interaction aggregation and the multiplex interaction aggregation. The former is to aggregate neighbors of the same interaction type and the latter is to combine the representations of different types.

5.2.1. Feature initialization

Before the message passing, we transform the initial nodes feature into the hidden representations as

$$\mathbf{H} = \tanh(\mathbf{X}\mathbf{W_0}). \tag{10}$$

Here $\mathbf{X}$ is the original attribute matrix of users and $\mathbf{W}_0 \in \mathbb{R}^{F\times d}$ is the linear transformation matrix, where F is the number of user attributes and d is the dimension of hidden representation.

5.2.2. User interaction aggregation

Given the ego graph $\mathcal{G}(u)$ with the ego user u and the interaction type t, we could obtain its l-order edge sets $\mathcal{E}_{u,t}^{(l)}$ and the l-order neighbor sets $\mathcal{N}_{u,t}^{(l)}$. The purpose of user interaction aggregation is to obtain the user representation under t interaction by iteratively aggregating the features of the l-order neighbors (the source users) into $(l-1)$-order neighbors (the destination users).

As each user may interact with many users, the contribution of different interactions cannot be specified artificially and should be learned by the model itself. As a result, we use the attention mechanism into consideration to learn each neighbor's coefficient in each layer, namely $\alpha_{(src,dst),t}$. With the learned coefficients normalized by softmax function across all neighbors, the personalized importance of src to user dst is automatically measured. Then dst's representation is updated by the attentive propagation of neighbor features. Formally, the updated representation of dst in view of interaction t can be calculated as

$$\mathbf{h}_{dst,t} = \sum_{(src,dst)\in\mathcal{E}_{u,t}^{(l)}} \alpha_{(src,dst),t}\mathbf{W}^{(l)}\mathbf{h}_{src}, \tag{11}$$

where $\alpha_{(src,dst),t}$ is the computed importance score of the interaction (src, dst), $\mathbf{W}^{(l)}$ is the transformation matrix for the l-order neighbors and $\mathbf{h}_{src}$ denotes the corresponding row of src in the user representation matrix $\mathbf{H}$.

Since the constructed ego graphs contain two different components, i.e. the seed interactions and the kNN co-interactions, we apply different operations to obtain the attention coefficients.

The attention on kNN co-interactions. In the calculation of $\alpha_{(src,dst),t}$ for the kNN co-interactions, we takes the source, destination and ego user into consideration, i.e. src, dst and u respectively. Concretely, the features of the source node and destination node of each edge are concatenated with the ego nodes in the ego graph, and fed into a fully connection layer followed by a non-linear activation. Then the importance score between the edge (src, dst) is obtained as

$$s_{(src,dst),t} = \mathbf{a}_t^{(l)\top}[\mathbf{W}^{(l)}\mathbf{h}_u||\mathbf{W}^{(l)}\mathbf{h}_{src}||\mathbf{W}^{(l)}\mathbf{h}_{dst}], \tag{12}$$

where $\mathbf{a}_t^{(l)} \in \mathbb{R}^{3d}$. Note that the features of ego user u are utilized in the calculation of each edge attention. It is because the kNN co-interactions are personalized with regard to user u, hence it is beneficial to consider u's features when assessing the importance of co-interactions. With the importance scores computed, we multiply them with the kNN distance of each co-interaction, and apply the softmax function across the incoming edges of dst to obtain the final coefficients, i.e.,

$$\alpha_{(src,dst),t} = \frac{\exp\left(\sigma\left(s_{(src,dst),t}\cdot k_{(src,dst),t}\right)\right)}{\sum_{(j,dst)\in\mathcal{E}_{u,t}^{(l)}}\exp\left(\sigma\left(s_{(j,dst),t}\cdot k_{(j,dst),t}\right)\right)}, \tag{13}$$

where $k_{(src,dst),t}$ is the kNN distance of the co-interaction (src, dst), which is recorded as edge features in the construction of ego graphs.

The attention on seed interactions. The calculation of $\alpha_{(src,dst),t}$ for the seed interactions is the simplified version of that on the co-interactions. There are only two users in the computation of attention, as dst user is the ego user on the seed interactions. Besides, the kNN distance is also removed as there is no such edge features for the seed interactions. The formulas are stated as

$$\alpha_{(src,u),t} = \frac{\exp\left(\sigma\left(\mathbf{a}_t^{(1)\top}[\mathbf{W}^{(1)}\mathbf{h}_{src}||\mathbf{W}^{(1)}\mathbf{h}_u]\right)\right)}{\sum_{(j,u)\in\mathcal{E}_{u,t}^{(1)}}\exp\left(\sigma\left(\mathbf{a}_t^{(1)\top}[\mathbf{W}^{(1)}\mathbf{h}_j||\mathbf{W}^{(1)}\mathbf{h}_u]\right)\right)}, \tag{14}$$

where $\mathbf{a}_t^{(1)} \in \mathbb{R}^{2d}$ is the importance vector in the first layer.

5.2.3. Multiplex interaction aggregation

Given the multiplex ego graph $\mathcal{G}(u)$, we could obtain T interaction-specific embeddings for the destination nodes, denoted as $\{\mathbf{h}_{dst,t}|t \in \mathcal{T}_E\}$. Generally, each user has conducted multiple types of interactions and it is intuitive that the user embeddings learned from various behaviors reflect different aspects of users' portraits. In order to measure the contributions of different interaction types for each user, another attention mechanism is applied. The interaction-specific embeddings are firstly transformed by a weight matrix, then the dot product between the trainable vector $\mathbf{q}_A$ and the transformed embeddings can be interpreted as the importance of interaction types, i.e.,

$$e_{dst,t} = \mathbf{q}_A^\top \tanh(\mathbf{A}\mathbf{h}_{dst,t} + \mathbf{b}), \tag{15}$$

$$\alpha_{dst,t} = \frac{\exp(e_{dst,t})}{\sum_{j\in\mathcal{T}_E}\exp(e_{dst,j})}, \tag{16}$$

where $\mathbf{A} \in \mathbb{R}^{d\times d}$ and $\mathbf{b} \in \mathbb{R}^d$ are the learnable parameters, with $\mathbf{q}_A^\top \in \mathbb{R}^d$ being the parameterized importance vector.

Once $\alpha_{dst,t}$ are computed for user u and all its interaction types, we can obtain the comprehensive representation by fusing the interaction-specific embeddings as

$$\mathbf{h}_{dst} = \sum_{t\in\mathcal{T}_E}\alpha_{dst,t}\mathbf{h}_{dst,t}. \tag{17}$$

5.2.4. Aggregation to the ego nodes

Now we have introduced basic MIGNN-layer, which perform the aggregation from the l-order neighbors to the $(l-1)$-order neighbors. It is noted that by iteratively replacing the (l) with $(l-1)$ from Eq. (11)–(17), the ego node u could receive the aggregation information up to its l-order neighbors, denoted as $\mathbf{h}_u^{(l)}$.

To explore the information in the ego graph from different orders, we perform the procedures of message aggregation for its $1, 2, \ldots, L$-order neighbors to obtain multiple representations for the ego node u, namely $\{\mathbf{h}_u^{(1)}, \mathbf{h}_u^{(2)}, \ldots, \mathbf{h}_u^{(L)}\}$. As these representations emphasize the connectivity of different orders, the final embeddings are generated as

$$\mathbf{h}_u^* = \sum_{l=0}^{L}\mathbf{h}_u^{(l)}. \tag{18}$$

In our experiments, we do not design special components for the layer combinations, as the uniform weighting of different order information leads to a good performance in general [28].

To summarize, the MIGNN module takes the multiplex ego graph as input and generates user embeddings for each ego user by attentively aggregating the features from neighbors of different orders.

5.3. Reciprocal information fusion

With three ego graphs augmented, we apply three independent MIGNN modules on these ego graphs as follows:

$$\mathbf{e}_{u,sim}^* = \mathbf{MIGNN}_{sim}(\mathcal{G}_{sim}, \mathbf{x}_u). \tag{19}$$

$$\mathbf{e}_{u,pref}^* = \mathbf{MIGNN}_{pref}(\mathcal{G}_{pref}, \mathbf{x}_u), \tag{20}$$

$$\mathbf{e}_{u,attr}^* = \mathbf{MIGNN}_{attr}(\mathcal{G}_{attr}, \mathbf{x}_u), \tag{21}$$

As is analyzed in Section 4.2, the learned embeddings $\mathbf{e}_{u,sim}^*$, $\mathbf{e}_{u,pref}^*$ and $\mathbf{e}_{u,attr}^*$ and suggest the information of user's similarity, preference and attraction, respectively.

The final step is to integrate the user embeddings learned from distinct aspects into the single comprehensive embeddings. We propose two variants as $GraphRR_{attn}$, and $GraphRR_{MLP}$, to examine these two candidate fusion functions:

- Attention mechanism: a shared transformation is applied to calculate the personalized importance of embeddings from different aspects, and the ultimate embedding is obtained by the weighted sum:

$$\begin{bmatrix}\alpha_{sim}\\ \alpha_{pref}\\ \alpha_{attr}\end{bmatrix}^\top = \mathrm{softmax}(\mathbf{q}_f \tanh(\mathbf{W}_f\begin{bmatrix}\mathbf{e}_{u,sim}^*\\ \mathbf{e}_{u,pref}^*\\ \mathbf{e}_{u,attr}^*\end{bmatrix}^\top + \mathbf{b}_f)) \tag{22}$$

$$\mathbf{e}_u^* = \alpha_{sim}\mathbf{e}_{u,sim}^* + \alpha_{pref}\mathbf{e}_{u,pref}^* + \alpha_{attr}\mathbf{e}_{u,attr}^*, \tag{23}$$

  where $\mathbf{W}_f \in \mathbb{R}^{d\times d}$, $\mathbf{b}_f \in \mathbb{R}^{d\times 1}$ and $\mathbf{q}_f \in \mathbb{R}^{1\times d}$ are the trainable parameters.
- Multi-Layer Perceptron (MLP): a two-layer MLP is applied for reciprocal information fusion.

$$\mathbf{e}_u^* = \mathbf{MLP}([\mathbf{e}_{u,sim}^*||\mathbf{e}_{u,pref}^*||\mathbf{e}_{u,attr}^*]). \tag{24}$$

5.4. Prediction

Given the ultimate embeddings generated from the reciprocal fusion, we estimate the reciprocity score between two users by inner product:

$$\hat{y}_{u,v} = \mathbf{e}_u^{*\top}\mathbf{e}_v. \tag{25}$$

To train the model in an end-to-end manner, we optimize the parameters in the model by minimizing the cross entropy via back propagation. The loss function is formulated as follows:

$$\mathcal{L} = -\sum_{(u,v)\in\Omega}\log\sigma(\hat{y}_{u,v}) - \sum_{(u',v')\in\Omega^-}\log(1-\sigma(\hat{y}_{u',v'})), \tag{26}$$

where $\sigma(\cdot)$ is the sigmoid function, Ω denotes the user pairs that fulfill reciprocity and Ω^- denotes the pairs that only indicate unilateral preferences or no subsequent interactions.

Table 2
Descriptive statistics of two user interaction graphs.

Graph	Type of interactions	# Interactions	# Nodes	# Features
KO	Approve Friend Request	0.97M	0.43M	47
	Accept Team Invitation	2.31M		
	Game Like	6.31M		
	Send Gift	0.33M		
	Send Message	0.51M		
T&J	Approve Friend Request	0.37M	0.31M	73
	Accept Team Invitation	0.41M		
	Send Message	0.49M		

5.5. Training and efficiency

We train GraphRR in the mini-batch setting for the large-scale datasets using multiple GPUs. The user features and the historical interaction edges are placed in CPU memory. The mini-batch contains the user pairs which have or do not have the reciprocity behaviors. We sample their interactions into the ego graphs which are loaded the graphs into GPU during training. For the ego graph augmentation, it can be implemented within the process of neighborhood sampling, and the kNN algorithm is preprocessed for each user and serialized into disks. Thus the per-batch space and time complexity of augmentation are fixed at $O(\prod_{i=1}^{L} S_i)$, where S_i is the sampling size of the ith layer.

6. Experiments

6.1. Experiment settings

6.1.1. User interaction graphs

In the experiments, we collect two user interaction graphs from two mobile games of various genres from *NetEase Games*,[2] which is one of the leading providers of game service to worldwide users. The multiplex edges of the graphs are built by their historical interactions. The attributes of nodes in each graph contain users' basic personal information, social activities and competitive performances from their historical records. The descriptive statistics of these two multiplex graphs are shown in Table 2 with more details of the graphs as follows:

- **KO**: The user interaction graph is collected from *Knives Out*,[3] a popular first-person shooting game. 47 user attributes are extracted and five types of user interactions occurring between April 12th, 2020 and May 11th, 2020 are recorded.
- **T&J**: This dataset is collected from *Tom and Jerry: Chase*,[4] a popular casual battle game. Similarly, 73 user attributes are extracted and the graph is constructed by three user interactions occurring between June 10th, 2020 and June 16th, 2020.

Besides, we show some examples of the user features used in the user interaction graph of *Knives Out* in Table 3. It is observed that the user features can be divided into four categories, which reveal user basic information, competitive ability, total activity and social engagement, respectively.

6.1.2. Datasets

There are four datasets in the experiments, which are collected following the period of interaction graph and shown in Table 4. These datasets describe whether users send *Friend Request* and *Team Invitation* to other users from the recommendation list after each game. The positive and the negative samples are both generated by users' explicit interactions.

As is shown in Fig. 2, the user interactions have an extremely skewed distribution in our scenario, i.e., around 1% active users contribute around 99% labels (interactions). If we directly use all labels for training, the labels of active users will dominate the overall optimization of the model. However, our RRS needs to provide the recommendation results not only for active users but also for the large proportion of long-tail inactive users. Hence, we apply the strategy of downsampling on the samples of active users to enforce the model to concentrate on inactive users. Specifically, we only keep n_{pos} positive labels and n_{neg} negative labels per user in each dataset, where n_{pos} and n_{neg} are the median size of positive and negative labels of all users.

6.1.3. Baselines

We compare GraphRR with some representative methods, including conventional RS models widely used in the industry (MLP, XGBoost), convectional RRS model (LFRR), homogeneous RS models (GAT, GraphSAGE, NGCF, LightGCN), and multiplex RS models (RGCN, HGT, MF-BPR and MBGCN):

Conventional Models:

- **MLP + DW** [36]: Multi-Layer Perceptron is a basic deep neural network to capture the complex feature dependencies. For fair comparisons with the graph-based methods, DeepWalk is also adopted to generate pre-trained node embeddings as the additional inputs for the MLP.
- **XGBoost + DW** [37]: XGBoost is a scalable gradient boosting framework and widely used in the industry. Similarly, DeepWalk is adopted as the additional feature extractor for fairness.
- **LFRR + DW** [10]: LFRR is a state-of-the-art method for reciprocal recommendation. It learns two latent factor models to calculate two preference scores, and obtain the final predicted score by combining them with aggregation operators, e.g. arithmetic mean. As it is not designed for capturing high-order user interactions, DeepWalk is adopted as the additional feature extractor.

Homogeneous Models:

- **GAT** [35]: It is a graph neural network method that considers the different contributions of neighbors by attention mechanism.
- **GraphSAGE** [38]: It is a graph neural network method for large-scale graphs and learns node representations by neighborhood sampling and feature aggregation.
- **NGCF** [11]: It is a state-of-the-art method in recommender systems that encodes the collaborative signal into node representations by modeling high-order connectivity in the user–item interaction graph.
- **LightGCN** [28]: It is a state-of-the-art method in recommender systems that simplifies the design of GCN by removing the feature transformation and nonlinear activation.

Multiplex Models:

- **RGCN** [39]: It is a relational graph neural network method that can be applied on the multiplex graph. It introduces interaction-specific transformations and combines the embeddings from different relations by the summing operator.
- **HGT** [29]: It is a graph representation method designed for web-scale heterogeneous graph. It introduces the node- and edge-type dependent parameters to calculate the heterogeneous attention over each edge.

2 https://www.neteasegames.com
3 https://www.knivesout-en.com
4 https://www.tomandjerrychaseasia.com

Table 3
The examples of user features from *Knives Out.*

Feature categories	Feature examples	Explanations
Basic information	*age*	The user's age
	region	The user's region
Competitive ability	*avg_single_kill*	The user's average kills per game in the past periods
	avg_single_life_time	The user's average survival time per game in the past periods
Total activity	*online_time*	The user's total online time in the past periods
	activity_days	The number of days the user is active in the past periods
Social engagement	*mic_cnt*	The number of games where the gamers has turned on the microphone to communicate in the past periods
	total_send_like	The number of Likes sent by the user in the past periods

Table 4
Descriptive statistics of the four datasets.

Datasets	KO-Invitation	KO-Request	T&J-Invitation	T&J-Request
Period	2 weeks	2 weeks	3 days	3 days
# users	0.32M	0.32M	0.19M	0.19M
# pos	1.09M	0.27M	0.63M	0.21M
# neg	0.75M	0.51M	4.01M	0.57M

- **MF-BPR** [22]: It is one of the popular models that utilize multi-behavior interaction data in recommender systems. It adapts the sampling strategy and applies pairwise loss to discriminate the strength of user behaviors.
- **MBGCN** [24]: It is a state-of-the-art GNN-based model for multi-behavior recommendation. It leverages the graph convolutional network to perform interaction-aware embedding propagation.

The proposed method and the baselines are implemented based on PyTorch and DGL [40]. For DW, GAT, GraphSAGE, NGCF, and LightGCN, all types of interactions are treated equally as they cannot deal with multiplex interactions, and they also neglect the directions of the user interaction graph. For GNN-based solutions, the depth of graph neural networks is set as 3. The dimensions of user representations and hidden layer are both 128, with the number of attention heads being 4. During training, Adam [41] is employed as the optimizer and the learning rate is set as 0.001.

6.2. Experimental results

6.2.1. Friend recommendation

To evaluate the performance of the proposed GraphRR in the task of friend recommendation, we conduct experiments in four datasets. The experiments are measured by two popular ranking metrics, i.e., Normalized Discounted Cumulative Gain (NDCG) and Mean Reciprocal Rank (MRR).

Specifically, for user u, the calculation of NDCG is formulated as

$$DCG_p = \sum_{i}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}, \tag{27}$$

$$NDCG_p = \frac{DCG_p}{IDCG_p}, \tag{28}$$

where i represents the ith recommendation candidate user for u. The total number of the candidates is denoted as p, and rel_i represents the relevance score between user i and u. Given user u, GraphRR ranks u's candidate users based on their relevance scores with u. Then DCG score is calculated to evaluate the quality of ranking of all candidates, and the Eq. (28) normalizes DCG score by the Ideal DCG (IDCG) which is the DCG score of ground truth ranking.

In terms of MRR, for user u, its MRR score is calculated as

$$MRR = \frac{1}{rank_i}, \tag{29}$$

where $rank_i$ represent the ranking of first positive candidate user for u.

The performance of GraphRR is illustrated in Table 5, where the relative improvements regarding the best baseline are recorded in the last row. We summarize some major observations as follows:

- Comparing LFRR + DW with MLP + DW and XGB + DW, we could find that considering reciprocity can improve models' performance for the recommendation. However, these methods perform worse than GNN-based methods as they could only separately model user attributes and interactions. It suggests the effectiveness of GNNs in integrating attributes and structures into embeddings.
- Methods that are aware of multiplexity, i.e. RGCN, HGT and GraphRR, can achieve better performance than those designed for homogeneous graphs, which further indicates the necessity of distinguishing various interactions in the multiplex user interaction graphs.
- Methods that are designed for the multi-behavior recommendation, i.e. MBGCN, and MF-BPR, can achieve superior performance than single-behavior methods such as LightGCN and NGCF. Besides, MBGCN which also uses the GNN architecture is the best multi-behavior baseline and has comparable results with the multiplex GNN-based methods (RGCN and HGT). This comparison verifies the effectiveness of GNN for capturing diverse user semantics for the multi-behavior recommendation.
- Our proposed method GraphRR consistently outperforms baselines in all metrics. In contrast to the multiplex methods HGT, RGCN, MF-BPR and MBGCN that ignore the unidirectionality of interactions, GraphRR can further improve the recommendation results by the ego graph augmentation to consider the reciprocity patterns between users.
- GraphRR_{MLP} performs generally better than GraphRR_{attn}. The reciprocal information fusing of GraphRR_{attn} is actually an additive operation with trainable coefficients, while GraphRR_{MLP} could explore the non-linear transformation and fully fuse the representations. The superiority of GraphRR_{MLP} verifies there might exist some complex, non-linear dependencies between users' representations on similarity, preference and attraction.
- The improvement of GraphRR on the **T&J** graph is more significant than that on the **KO** graph. This may be caused by the distinction between these two games' competitive contents. In *Tom and Jerry: Chase*, a gamer has a relatively fixed role, and the pattern of cooperation between teammates is also simpler, whereas *Knives Out* is more competitive and users' gaming styles are more uncertain, rendering the task of reciprocal recommendation on the **KO** graph more challenging.

Table 5
The ranking performance of different methods for the recommendation task with the best results in each datasets marked in bold and the best baseline underlined.

Datasets	KO-Invitation		KO-Request		T&J-Invitation		T&J-Request	
Metric	NDCG	MRR	NDCG	MRR	NDCG	MRR	NDCG	MRR
MLP + DW	0.5770	0.7705	0.6943	0.8225	0.5382	0.7534	0.7421	0.8734
XGB + DW	0.5931	0.7631	0.6938	0.8326	0.5676	0.7957	0.7294	0.8439
LFRR + DW	0.6026	0.7695	0.7081	0.8372	0.6148	0.8329	0.7508	0.8856
GAT	0.6089	0.7521	0.7047	0.8281	0.5563	0.7899	0.7635	0.9251
GraphSAGE	0.6056	0.7699	0.7151	0.8398	0.5679	0.7960	0.7870	0.9346
NGCF	0.6034	0.7709	0.7128	0.8551	0.6008	0.8179	0.7843	0.9367
LightGCN	0.6252	<u>0.8058</u>	0.7041	0.8366	0.6171	0.8068	0.7747	0.9304
RGCN	<u>0.6330</u>	0.7973	0.7314	0.8502	0.6503	0.8459	0.7877	<u>0.9477</u>
HGT	0.6326	0.7849	0.7403	<u>0.8686</u>	0.6261	0.8244	<u>0.8010</u>	0.9346
MF-BPR	0.6134	0.7701	0.7121	0.8234	0.6029	0.7992	0.7495	0.8802
MBGCN	0.6321	0.8013	<u>0.7425</u>	0.8637	<u>0.6654</u>	<u>0.8493</u>	0.7885	0.9424
GraphRR$_{attn}$	0.6423	0.8029	0.7533	**0.8936**	0.6931	0.9045	0.8124	0.9431
GraphRR$_{MLP}$	**0.6483**	**0.8187**	**0.7542**	0.8907	**0.7255**	**0.9266**	**0.8311**	**0.9582**
Impr%	2.42%	1.60%	1.58%	2.88%	9.03%	9.10%	3.76%	1.11%
p-value	2.20E−3	1.01E−3	4.29E−3	1.71E−4	2.17E−6	4.58E−8	7.02E−6	1.65E−2

To further compare the statistical robustness of the improvement of the proposed GraphRR over baseline methods, we calculate confidence intervals with the ten runs and report the p-values of a paired t-test between GraphRR$_{MLP}$ and one of the best baseline MBGCN. As shown in Table 5, the improvements of GraphRR over MBGCN are statistically significant with the paired t-test at $p < 0.05$ on all evaluation metrics over all datasets. Overall, the experimental results demonstrate the effectiveness of GraphRR.

6.2.2. Reciprocity evaluation

The reciprocity between users is evaluated by two labels generated from the existing users' responses to the preference initiated by others, i.e. **Co-gaming** and **Friending**, which indicates whether users *Accept Team Invitation* or *Approve Friend Request* with the recommended users, respectively. We choose these two labels for reciprocity evaluation as they are important and challenging for online games. For each user, the positive and negative labels are collected from their responses to the recommendation results.

Group AUC (GAUC) [42] is adopted as the metric, which measures the goodness of intra-user ranking and is shown to be more relevant to online performance in the recommender system. Similar to [42], we introduce RelaImpr to measure the relative AUC improvement over the models. Since a random strategy present an AUC of 0.5, RelaImpr is defined as

$$RelaImpr = \left(\frac{AUC(measured\ model) - 0.5}{AUC(base\ model) - 0.5} - 1\right) * 100\%. \tag{30}$$

The results of reciprocity evaluation are shown in Table 6. We also report the p-values of a paired t-test between GraphRR$_{MLP}$ and one of the best baseline RGCN. We observe consistently high gains for the two variants of GraphRR on both datasets that reflect users' reciprocity, where the improvements are statistically significant with a paired t-test at $p < 0.05$. These results demonstrate the effectiveness of the proposed ego graph augmentation to capture users' reciprocity patterns.

6.3. Model analysis

6.3.1. Ablation study

To gain insights into GraphRR's architecture, we study the impact of its significant components. Concretely, we compare GraphRR with its three variants: GraphRR$_{-aug}$, GraphRR$_{-co}$ and GraphRR$_{-multi}$, which are defined as follows:

- GraphRR$_{-aug}$: The proposed ego graph augmentation is removed. This variant only utilizes the similarity-based ego graph to learn user representations, which neglects the directions of the original user interactions.
- GraphRR$_{-co}$: The kNN co-interaction graph in the reciprocity-based ego graph augmentation is removed, where the preference and attraction ego graphs still exist but only have a single hop neighborhood.
- GraphRR$_{-multi}$: This variant ignores the multiplexity of the user interaction graph, i.e., all types of interactions have the same contribution to users' representations.

The comparisons of GraphRR with the above three variants are illustrated in Fig. 8. We can observe that missing any of the components leads to performance degeneration of GraphRR. There are some other specific conclusions:

- The improvements of GraphRR over GraphRR$_{-aug}$ suggests a notable gain from the proposed ego graph augmentation, which verifies GraphRR's effectiveness to deal with user interactions of preference and attraction.
- GraphRR$_{-co}$ is marginally better over GraphRR$_{-aug}$, which indicates that separately learning the attraction and preference components from the user interaction graph can slightly contribute to the model. Meanwhile, the kNN co-interaction graph plays a more crucial role in the ego graph augmentation comparing GraphRR with GraphRR$_{-co}$.
- The superiority of GraphRR over GraphRR$_{-multi}$ shows that distinguishing multiplex user interactions makes noteworthy improvements, and confirms that the users' multiplex interactions are essential to estimate the reciprocity between users.

6.3.2. Effect of the ego graph augmentation (**RQ**$_4$)

To further verify the efficacy of the core component of the proposed architecture, i.e. the ego graph augmentation, we facilitate the baseline methods with this mechanism and record the relative improvement in Table 7. Specifically, we keep the ego graph augmentation and the reciprocal information fusion, and only replace the MIGNN modules with the baseline methods.

The results demonstrate that the proposed ego graph augmentation generally brings about improvement on the baselines to a varying extent. Concretely, the benefits of the ego graph augmentation on **T&J** graph are more significant compared to those on **KO** graph, which is consistent with the results in Table 5. We think this is caused by the fact that the reciprocity pattern between gamers is more evident in *Tom and Jerry: Chase*.

Table 6
The experimental results on two reciprocity datasets.

Dataset	KO-Invitation	KO-Request	T&J-Invitation	T&J-Request
Reciprocity indicator	Co-gaming	Friending	Co-gaming	Friending
MLP+DW	0.5936	0.6489	0.5449	0.5597
XGB+DW	0.5920	0.6445	0.5581	0.5524
LFRR+DW	0.6068	0.6531	0.5629	0.5684
GAT	0.6020	0.6454	0.5608	0.5898
GraphSAGE	0.6133	0.6639	0.5789	0.6100
NGCF	0.5976	0.6527	0.6068	0.6139
LightGCN	0.6096	0.6646	0.5899	0.5949
RGCN	<u>0.6248</u>	0.6730	<u>0.6555</u>	<u>0.6345</u>
HGT	0.6179	<u>0.6797</u>	0.6239	0.6215
MF-BPR	0.5994	0.6207	0.5660	0.5893
MBGCN	0.6201	0.6753	0.6542	0.6259
GraphRR$_{attn}$	**0.6411**	0.7186	0.6869	0.6530
GraphRR$_{MLP}$	0.6403	**0.7224**	**0.6946**	**0.6788**
RelaImpr	13.06%	23.76%	25.11%	32.94%
p-value	1.53E−2	8.98E−7	7.45E−6	3.06E−7

Table 7
The performance of the baseline methods facilitated by the augmented ego graphs.

Datasets	Methods	Invitation		Request	
		NDCG	MRR	NDCG	MRR
KO	NGCF$_{aug}$	+3.51%	+2.79%	+2.43%	−0.57%
	LightGCN$_{aug}$	+1.28%	+1.01%	+3.03%	+3.49%
	RGCN$_{aug}$	+2.24%	+2.18%	+2.61%	+2.38%
	HGT$_{aug}$	+1.56%	+1.94%	+1.50%	+1.92%
T&J	NGCF$_{aug}$	+4.45%	+0.26%	+0.05%	+0.68%
	LightGCN$_{aug}$	+4.77%	+0.33%	+1.28%	+1.81%
	RGCN$_{aug}$	+11.37%	+8.46%	+6.47%	+2.18%
	HGT$_{aug}$	+13.40%	+10.49%	+3.55%	+0.95%

Table 8
The learned coefficients (mean values) for the multiplex interactions.

Interaction	KO-Invitation	KO-Request	T&J-Invitation	T&J-Request
Approve Friend Request	0.12	0.39	0.28	0.39
Accept Team Invitation	0.44	0.13	0.43	0.31
Game Like	0.21	0.28	\	\
Send Gift	0.10	0.09	\	\
Send Message	0.13	0.11	0.29	0.30

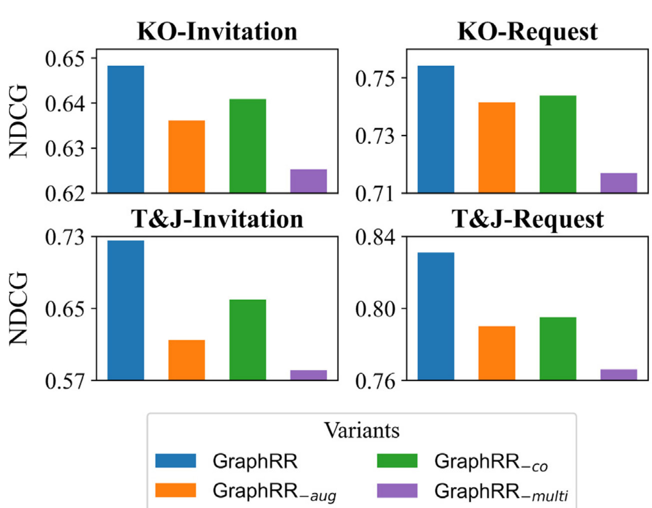


Fig. 8. The performances of GraphRR and its three variants.

Meanwhile, we also observe that the multiplexity-aware methods present more improvements than the homogeneous methods on **T&J** graph. It suggests that the joint consideration of multiplexity and reciprocity significantly boosts the model's performance.

*6.3.3. The analysis of the multiplexity attention (**RQ$_2$**)*

In this section, we investigate the contribution of the multiplex interactions in the graph. We extract the attention coefficients and record them in Table 8. It can be observed that *Accept Team Invitation* and *Approve Friend Request* accounts for a high proportion of contributions to the labels **Invitation** and **Request** respectively, which is consistent with the observation in reality that the most relevant interactions provide a primary contribution to predictions while other interactions only play the auxiliary roles.

6.3.4. Hyper-parameter studies

To understand how hyper-parameters influence the performance of the proposed GraphRR, we conduct the sensitivity studies on several important hyper-parameters including the number of co-interactions k in the ego graph augmentation, the number of GNN layers, and the dimension of the hidden layer. We record the performance of GraphRR$_{MLP}$ in Fig. 9. The observations are summarized as follows:

- **The number of co-interactions** k. In this experiment, the number of co-interactions k is searched from 0 to 30. We can observe that with the increase of k, the performance raise first and then start to drop slightly. It is reasonable as k controls the homophily of the augmented ego graph. When k is small, there are no enough neighbors in the ego-graph, thus the augmentation has little impact on the

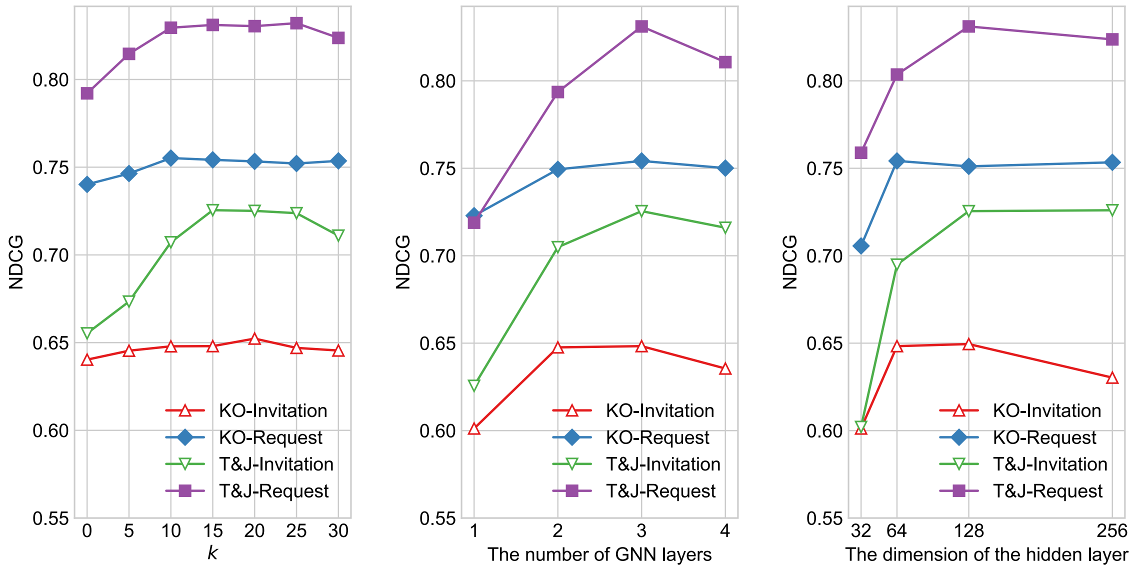


Fig. 9. The sensitivity of the hyper-parameters in GraphRR.

model. When k is large, the user's ego graph may contain the noisy interactions with heterophily, thus reducing the performance of the proposed MIGNN.

- **The number of GNN layers.** By stacking the proposed MIGNN layers, GraphRR could capture the complex high-order user semantics. As is observed in Fig. 9, we can notice that the 2-layer or 3-layer GraphRR performs significantly better than the 1-layer GraphRR, which suggests the positive effect of high-order user interactions. Meanwhile, the performance of the 4-layer GraphRR starts to deteriorate. Generally, three GNN layers are sufficient to capture the high-order signals, while more layers might introduce noise and lead to over-smoothing [43].
- **The dimension of the hidden layer.** Here we investigate the effect of the dimension of the hidden layer. Based on the results, we can see that the performances of GraphRR generally increase with the hidden dimension due to the larger representation feature space. However, large hidden dimension do not necessarily result in improvement, and GraphRR achieves the best performance when the dimension is set as 128. After that, it begins to degenerate. The reason is that GraphRR requires a suitable dimension to encode the user interactions, and a larger dimension brings a higher risk of overfitting.

7. Conclusion and future work

In this paper, we propose a multiplex Graph based Reciprocal friend Recommender system (GraphRR), which exploits the multiplex user interactions by graph neural networks in reciprocal recommendation scenarios.

GraphRR fully explores the reciprocity patterns between users by the reciprocity-based ego-graph augmentation, and captures users' rich behavioral semantics by attentive multiplex graph neural networks. Experiments on the datasets from two online games demonstrate the effectiveness of GraphRR and the proposed ego-graph augmentation.

For future work, we believe that the GNN-based solutions are one of the promising research directions for reciprocal recommender systems, as they have great capacities to capture the multiplex high-order relations between users. Hence it remains a challenge to design a suitable GNN architecture for recommender systems. Besides, as the consideration of reciprocity can significantly improve the model's performance in the reciprocal recommendation, the future exploration of the reciprocity patterns can also go beyond the techniques of graph augmentation.

CRediT authorship contribution statement

Yaomin Chang: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation; Writing – original draft, Writing – review & editing, Visualization. **Lin Shu:** Conceptualization, Methodology, Writing – original draft. **Erxin Du:** Methodology, Software, Validation, Writing – original draft. **Chuan Chen:** Conceptualization, Methodology, Writing – review & editing, Supervision. **Ziyang Zhang:** Conceptualization, Methodology, Writing – review & editing. **Zibin Zheng:** Resources, Project administration. **Yuzhao Huang:** Validation, Writing – review & editing. **Xingxing Xing:** Data curation.

Declaration of competing interest

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

Acknowledgments

The research is supported by the National Natural Science Foundation of China 62176269 62032025, the Guangdong Basic and Applied Basic Research Foundation, China 2019A1515011043, 2018B030312002.

References

[1] W. Chen, P. Huang, J. Xu, X. Guo, C. Guo, F. Sun, C. Li, A. Pfadler, H. Zhao, B. Zhao, POG: personalized outfit generation for fashion recommendation at Alibaba iFashion, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2662–2670.
[2] L. Chen, Y. Liu, X. He, L. Gao, Z. Zheng, Matching user with item set: Collaborative bundle recommendation with deep attention network., in: IJCAI, 2019, pp. 2095–2101.
[3] R. Ying, R. He, K. Chen, P. Eksombatchai, W.L. Hamilton, J. Leskovec, Graph convolutional neural networks for web-scale recommender systems, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 974–983.

[4] P. Covington, J. Adams, E. Sargin, Deep neural networks for youtube recommendations, in: Proceedings of the 10th ACM Conference on Recommender Systems, 2016, pp. 191–198.
[5] J. Neve, I. Palomares, Hybrid reciprocal recommender systems: Integrating item-to-user principles in reciprocal recommendation, in: Companion Proceedings of the Web Conference 2020, 2020, pp. 848–853.
[6] L. Pizzato, T. Rej, T. Chung, I. Koprinska, J. Kay, RECON: a reciprocal recommender for online dating, in: Proceedings of the Fourth ACM Conference on Recommender Systems, 2010, pp. 207–214.
[7] L. Li, T. Li, MEET: a generalized framework for reciprocal recommender systems, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, 2012, pp. 35–44.
[8] P. Xia, B. Liu, Y. Sun, C. Chen, Reciprocal recommendation system for online dating, in: 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM, IEEE, 2015, pp. 234–241.
[9] A. Kleinerman, A. Rosenfeld, F. Ricci, S. Kraus, Optimally balancing receiver and recommended users' importance in reciprocal recommender systems, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 131–139.
[10] J. Neve, I. Palomares, Latent factor models and aggregation operators for collaborative filtering in reciprocal recommender systems, in: Proceedings of the 13th ACM Conference on Recommender Systems, 2019, pp. 219–227.
[11] X. Wang, X. He, M. Wang, F. Feng, T.-S. Chua, Neural graph collaborative filtering, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 165–174.
[12] F. Xie, Z. Cao, Y. Xu, L. Chen, Z. Zheng, Graph neural network and multi-view learning based mobile application recommendation in heterogeneous graphs, in: 2020 IEEE International Conference on Services Computing, SCC, IEEE, 2020, pp. 100–107.
[13] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S.Y. Philip, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. (2020).
[14] J. Zhao, Z. Zhou, Z. Guan, W. Zhao, W. Ning, G. Qiu, X. He, Intentgc: a scalable graph convolution framework fusing heterogeneous information for recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2347–2357.
[15] C. Yang, A. Pal, A. Zhai, N. Pancha, J. Han, C. Rosenberg, J. Leskovec, MultiSage: Empowering GCN with contextualized multi-embeddings on web-scale multipartite networks, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 2434–2443.
[16] P. Kumar, R.S. Thakur, Recommendation system techniques and related issues: a survey, Int. J. Inf. Technol. 10 (4) (2018) 495–501.
[17] R. Chen, Q. Hua, Y.-S. Chang, B. Wang, L. Zhang, X. Kong, A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks, IEEE Access 6 (2018) 64301–64320.
[18] Y. Liu, C. Liang, X. He, J. Peng, Z. Zheng, J. Tang, Modelling high-order social relations for item recommendation, IEEE Trans. Knowl. Data Eng. (2020).
[19] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 173–182.
[20] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, T.-S. Chua, Nais: Neural attentive item similarity model for recommendation, IEEE Trans. Knowl. Data Eng. 30 (12) (2018) 2354–2366.
[21] Z. Zhao, Z. Cheng, L. Hong, E.H. Chi, Improving user topic interest profiles by behavior factorization, in: Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 1406–1416.
[22] B. Loni, R. Pagano, M. Larson, A. Hanjalic, BayesIan personalized ranking with multi-channel user feedback, in: Proceedings of the 10th ACM Conference on Recommender Systems, 2016, pp. 361–364.
[23] C. Gao, X. He, D. Gan, X. Chen, F. Feng, Y. Li, T.-S. Chua, D. Jin, Neural multi-task recommendation from multi-behavior data, in: 2019 IEEE 35th International Conference on Data Engineering, ICDE, IEEE, 2019, pp. 1554–1557.
[24] B. Jin, C. Gao, X. He, D. Jin, Y. Li, Multi-behavior recommendation with graph convolutional networks, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 659–668.
[25] I. Palomares, C. Porcel, L. Pizzato, I. Guy, E. Herrera-Viedma, Reciprocal recommender systems: Analysis of state-of-art literature, challenges and opportunities towards social recommendation, Inf. Fusion 69, 103–127.
[26] R.v.d. Berg, T.N. Kipf, M. Welling, Graph convolutional matrix completion, 2017, arXiv preprint arXiv:1706.02263.
[27] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.
[28] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: Simplifying and powering graph convolution network for recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 639–648.
[29] Z. Hu, Y. Dong, K. Wang, Y. Sun, Heterogeneous graph transformer, in: Proceedings of the Web Conference 2020, 2020, pp. 2704–2710.
[30] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, D. Koutra, Beyond homophily in graph neural networks: Current limitations and effective designs, Adv. Neural Inf. Process. Syst. 33 (2020).
[31] C. Zhou, Y. Liu, X. Liu, Z. Liu, J. Gao, Scalable graph embedding for asymmetric proximity, Proceedings of the AAAI Conference on Artificial Intelligence 31 (1) (2017).
[32] P. Cui, X. Wang, J. Pei, W. Zhu, A survey on network embedding, IEEE Trans. Knowl. Data Eng. 31 (5) (2018) 833–852.
[33] H. Nt, T. Maehara, Revisiting graph neural networks: All we have is low-pass filters, 2019, arXiv preprint arXiv:1905.09550.
[34] B. Xu, H. Shen, Q. Cao, K. Cen, X. Cheng, Graph convolutional networks using heat kernel for semi-supervised learning, 2020, arXiv preprint arXiv:2007.16002.
[35] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, 2017, arXiv preprint arXiv:1710.10903.
[36] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710.
[37] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[38] W.L. Hamilton, R. Ying, J. Leskovec, Inductive representation learning on large graphs, 2017, arXiv preprint arXiv:1706.02216.
[39] M. Schlichtkrull, T.N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: European Semantic Web Conference, Springer, 2018, pp. 593–607.
[40] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, Z. Zhang, Deep graph library: A graph-centric, highly-performant package for graph neural networks, 2019, arXiv preprint arXiv:1909.01315.
[41] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[42] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, K. Gai, Deep interest network for click-through rate prediction, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1059–1068.
[43] Q. Li, Z. Han, X.-M. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.