2022 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT) | 978-1-6654-9402-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/WI-IAT55865.2022.00016

Multiple Neighbor Relation Enhanced Graph Collaborative Filtering

Riwei Lai[1,2], Shitong Xiao[1], Rui Chen[*,1], Li Chen[2], Qilong Han[1], Li Li[3]
[1]*College of Computer Science and Technology, Harbin Engineering University*
[2]*Department of Computer Science, Hong Kong Baptist University*
[3]*Department of Electrical and Computer Engineering, University of Delaware*
{lai, ruichen, hanqilong}@hrbeu.edu.cn, shitong479@gmail.com, lichen@comp.hkbu.edu.hk, lilee@udel.edu

***Abstract*—Graph convolutional networks (GCNs) have substantially advanced state-of-the-art collaborative filtering (CF) methods. Recent GCN-based CF methods have started to explore potential neighbor relations instead of only focusing on direct user-item interactions. Despite the encouraging progress, they still suffer from two notable limitations: (1) only one type of potential neighbor relations is explored, i.e., co-interacting with the same item/user, neglecting the fact that user-item interactions are associated with various attributes and thus there can exist multiple potential neighbor relations from different aspects; (2) the distinction between information from direct user-item interactions and potential neighbor relations and their different extents of influence are not fully considered, which represent very different aspects of a user or an item. In this paper, we propose a novel Multiple Neighbor Relation enhanced method for Graph Collaborative Filtering (MNR-GCF) to address these two limitations. First, in order to capture multiple potential neighbor relations, we introduce a new construction of heterogeneous information networks with multiple types of edges to account for multiple neighbor relations, and a multi-relation aggregation mechanism to effectively integrate relation-aware information. We then enhance CF with a degree-aware dynamic routing mechanism to dynamically and adaptively fuse information from direct user-item interactions and potential neighbor relations at each aggregation layer. Our extensive experimental results show that our solution consistently and substantially outperforms a large number of state-of-the-art CF methods on three public benchmark datasets.**

***Index Terms*—Collaborative filtering, graph convolutional network, neighbor relation**

I. Introduction

Recommender systems have been widely adopted in different business applications, such as e-commerce [1], [2], online advertising [3], and social media platforms [4], to alleviate the issue of information overload, improve user experience and boost revenue. Collaborative filtering (CF) is a widely used recommendation technique that leverages collaborative information among users and items to predict users' preferences [5], [6].

Recently, graph-based methods, especially those based on graph convolutional networks (GCNs), have been increasingly adopted in CF. They formulate both users and items as nodes in a graph and allow to jointly investigate the correlations among users and items. Existing GCN-based works usually model user-item interactions as a bipartite graph [7], [8], where users and items are two disjoint node sets, and edges between them indicate their interactions. Thus, the high-order collaborative information between users and items can be captured by stacking multiple propagation layers on the user-item bipartite graph. However, the above methods only focus on mining users' preferences from the *direct user-item interactions*, which suffers from two major limitations. On the one hand, direct user-item interactions might be sparse for inactive users or unpopular items, resulting in the cold-start problem or the popularity bias issue, which impedes the quality of representations. On the other hand, due to the inherent limitation of a recommender system's exposure mechanism, the direct interacted items of a user may not be exhaustive to reflect all his/her potential preferences [6], [9]–[11]. As such, some very recent studies [6], [10]–[15] have started to explore potential neighbor relations in different ways to enhance CF. For example, Multi-GCCF [11] leverages potential user-user and item-item relations by calculating pairwise cosine similarities between users' or items' interactions. NIA-GCN [6] randomly selects K users/items co-interacted with the same item/user to construct potential neighbor relations.

Fig. 1. An illustration of user-item interactions in the e-commerce scenario. The interactions contain various attributes, such as timestamp, rating, and behavior, which are critical to reveal different potential neighbor relations.

While these latest efforts lead to encouraging results, we argue that there are still several problems to be considered in order to further enhance the model's performance. **First**, previous studies usually investigate only one type of potential neighbor relations, i.e., co-interacting with the same item/user,

Supported by the National Key R&D Program of China under Grant No. 2020YFB1710200, and the National Natural Science Foundation of China under Grant No.61872105 and No.62072136.
*Corresponding author

978-1-6654-9402-1/22/$31.00 ©2022 IEEE
DOI 10.1109/WI-IAT55865.2022.00016

despite the fact that user-item interactions can be associated with various attributes, such as timestamp, rating, and behavior (e.g., click, purchase, browse, etc.) [16], which leads to *multiple potential neighbor relations* from different aspects. Fig. 1 shows an example of the various attributes of user-item interactions in the e-commerce scenario. In this example, user A and user B both bought the quilted jacket in summer probably because the quilted jacket was on sale, reflecting the similarity of their purchasing habits (e.g., paying attention to products' price-performance ratios), while user A and user C share similar preferences on the quilted jacket because they gave the same rating. **Second**, most existing methods do not discriminate the information from direct user-item interactions and potential neighbor relations, which represent very different aspects of a user or an item [15]. The neighboring users of a target user can normally well unveil his/her general preference, while the interacted items of a target user may better indicate his/her specific preferences. To the best of our knowledge, Multi-GCCF [11] is the only work that discriminates the information from direct user-item interactions and potential neighbor relations using multiple graphs. However, such information is simply concatenated in *the final output layer* and treated *equally* without a careful distinction, which may lead to an incomplete understanding of user preferences. As such, the importance of information from direct user-item interactions and potential neighbor relations is different and needs to be more carefully factored in.

In addressing these aforementioned problems, we propose a **M**ultiple **N**eighbor **R**elation enhanced **G**raph **C**ollaborative **F**iltering (**MNR-GCF**), which extends information sources of a user or an item to its neighbors under multiple relations in terms of pre-defined **heterogeneous information networks** (HINs). Specifically, we construct the HINs based on various attributes of user-item interactions. Take the construction of a user-user heterogeneous information network as an example: A user-user heterogeneous information network consists of one type of nodes representing users and multiple types of edges, each of which corresponds to users' interactions with common items with respect to a specific attribute, e.g., timestamp, and represents a potential relation between users. An item-item heterogeneous information network is constructed in a similar way. After that, a **multi-relation aggregation mechanism** is proposed to aggregate relation-aware information from HINs. Note that the HINs enable to effectively capture the information from multiple neighbor relations, similar to existing works, we use a user-item bipartite graph to capture the information from direct user-item interactions. Finally, after obtaining the aggregated information from different sources, we design a **degree-aware dynamic routing mechanism** to *dynamically* and *adaptively* fuse them by considering the affinity between the fused information and the aggregated information from different sources at *each aggregation layer*.

We summarize our key contributions as follows.

- We highlight the importance of multiple neighbor relations and explicitly model them in terms of heterogeneous information networks by considering multiple attributes of user-item interactions. A multi-relation aggregation mechanism is introduced to aggregate relation-aware information.
- We design a simple yet effective degree-aware dynamic routing mechanism to fuse the aggregated information from different sources, in which node degrees are used to initialize the importance of different information sources. Unlike existing works, the dynamic routing mechanism enables dynamical and adaptive fusion at each aggregation layer.
- We perform extensive experiments on three public datasets and demonstrate that our proposed solution substantially and consistently outperforms a wide range of state-of-the-art competitors.

II. Related Work

A. Model-Based CF Methods

CF is a prevalent technique in modern recommender systems. The core of CF lies in how to design a model so that it can learn more representative embeddings from similar users or items. Earlier CF models like matrix factorization (MF) [17], [18] project users and items as embedding vectors and conduct inner product as an interaction function between them to predict an unobserved interaction. However, such an interaction function is insufficient to reveal the complex and nonlinear relations between users and items. For this reason, some recent works focus on exploiting deep learning techniques to enhance the interaction function. For instance, NeuMF [19] employs nonlinear neural networks as the interaction function. More recent works have found that different historical interactions contribute differently to the prediction of future interactions. To this end, attention mechanisms, such as ACF [20] and DeepICF [21], are introduced to automatically learn the importance of each historical interaction.

B. GCN-Based CF Methods

Recently, graph convolutional networks (GCNs) have attracted increasing attention for CF due to their powerful capability of capturing the collaborative information among users and items. Many GCN-based CF methods [7], [8], [22]–[27] have been developed. GC-MC [22] proposes a graph convolutional auto-encoder for explicit matrix completion. PinSage [23] adopts random walks on an item-item graph for image recommendation. NGCF [7] captures the high-order collaborative information between users and items by stacking multiple embedding propagation layers on a user-item graph. Inspired by the study on simplifying GCNs [28], researchers also introspect on the complex designs of GCN-based recommendation models. LightGCN [8] shows that transformation functions and nonlinear activations have limited positive effects on CF and sometimes might even degrade the performance. By removing these two components, it yields better performance on CF tasks. To alleviate the inherent over-smoothing problem of GCN, LR-GCCF [25] revisits the GCN-based CF methods with a linear residual graph convolutional approach. Due to the limitations of only mining users' preferences from direct user-item interactions, some very recent works [6], [11]–[15] have started to explore potential neighbor relations in different ways to enhance CF. Multi-GCCF [11] constructs two separate user-user and item-item graphs. It employs a multi-graph encoding layer to integrate the information provided by user-item, user-user and item-item graphs. NIA-GCN [6] develops a cross-depth ensemble layer to extract user-user, item-item and user-item relations from a single user-item graph. NGAT4Rec [29] employs a neighbor-aware graph attention layer that assigns different neighbor-aware attention coefficients to different neighbors of a given node to capture neighbor relations. EGLN [30] calculates the cosine similarity between users' and items' embeddings, and retains the edges with top-K computed similarities to build an enhanced user-item graph. Distinct from mainstream CF models that model interactions in a uniform manner, DGCF [31] and IMP [32] pay special attention to user-item relationships at a finer granularity of user intents. DGCF disentangles representations of users and items, while IMP decomposes a user-item graph as multiple subgraphs for message passing. Instead of using explicit message passing, UltraGCN [12] directly approximates the limit of infinite-layer graph convolutions via constraint losses for efficient recommendation. However, all the above GCN-based methods still suffer from the two notable problems discussed previously, which motivates our proposed solution.

III. Proposed Model

In this section, we present our proposed MNR-GCF framework in detail, whose architecture is illustrated in Fig. 2. There are five components in the framework: (1) a heterogeneous graph construction component that constructs heterogeneous information networks; (2) an embedding initialization component that initializes user embeddings and item embeddings; (3) multiple aggregation layers that refine the embeddings by aggregating information from direct user-item interactions and multiple neighbor relations; (4) a layer combination module that aggregates the refined embeddings from different aggregation layers; and (5) a prediction component that outputs the prediction score of a user-item pair.

A. Heterogeneous Graph Construction

Unlike existing methods that explore only one type of potential neighbor relations, we introduce multiple potential neighbor relations in terms of heterogeneous information networks (HINs) based on various attributes of user-item interactions. In what follows, we focus on explaining how to construct a user-user HIN. An item-item HIN can be constructed in a similar way.

The user-user HIN consists of one type of nodes representing users and multiple types of edges, where each type of edges corresponds to users' interactions with common items with respect to a specific attribute and represents one of the potential relations between users. For ease of presentation and better clarity, given a set of interaction attributes $\mathcal{A}$, we illustrate the idea by considering two example attributes, $\mathcal{A} = \{\text{timestamp}, \text{rating}\}$. These two selected attributes are present in various datasets, for example, the benchmark datasets we use in the experiments. We point out that *it is straightforward to include other types of attributes under our construction of HINs*. For timestamp, we consider the time interval between two interactions to create an edge between user u and user v who interacted with the same item i via

$$\mathbf{A}_{uv}^{\mathsf{U},\mathsf{timestamp}} = \sum_{i \in \mathcal{C}_{uv}} \mathbb{I}(|t_{ui} - t_{vi}| < \gamma). \tag{1}$$

Here $\mathbf{A}_{uv}^{\mathsf{U},\mathsf{timestamp}}$ is the element corresponding to user u and user v in the user-user HIN's adjacency matrix $\mathbf{A}^{\mathsf{U},\mathsf{timestamp}}$ by considering timestamp. $\mathcal{C}_{uv}$ is the set of common interacted items of user u and user v. t_{ui} is the timestamp of the interaction between user u and item i. $\mathbb{I}(\cdot)$ is an indicator function, which returns 1 when the condition holds and 0 otherwise. γ is a predefined threshold to filter out less reliable relations. The intuition of this design is that the shorter the time interval between two interactions with the same item is, the more similar the two users are, which is similar to the idea in TiSASRec [33].

For rating, we create an edge between user u and user v if they interacted with the same item and gave the same rating:

$$\mathbf{A}_{uv}^{\mathsf{U},\mathsf{rating}} = \sum_{i \in \mathcal{C}_{uv}} \mathbb{I}(r_{ui} = r_{vi}). \tag{2}$$

where $\mathbf{A}_{uv}^{\mathsf{U},\mathsf{rating}}$ is the corresponding element in the user-user HIN's adjacency matrix $\mathbf{A}^{\mathsf{U},\mathsf{rating}}$ by considering rating, r_{ui} is user u's rating on item i. The user-user and item-item HINs are illustrated in Fig. 2. Please note that they are weighted undirected graphs.

In addition to the novel user-user and item-item HINs, we follow existing works [7], [8] to model user-item interactions by constructing a user-item bipartite graph, where users and items are two disjoint node sets, and edges between them indicate their interactions.

B. Embedding Initialization

There are two types of nodes in the graphs, namely user nodes and item nodes. We use u and v to denote user nodes, and i and j to denote item nodes. Embedding initialization aims at mapping the IDs of user u and item i into dense embedding vectors $\mathbf{e}_u^{(0)} \in \mathbb{R}^d$ and $\mathbf{e}_i^{(0)} \in \mathbb{R}^d$, where d is the dimension of embedding vectors. We build two parameter matrices as embedding look-up tables for embedding initialization:

$$\mathbf{e}_u^{(0)} = \mathbf{P}^\top \mathbf{x}_u, \quad \mathbf{e}_i^{(0)} = \mathbf{Q}^\top \mathbf{x}_i, \tag{3}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are trainable parameter matrices of users and items, and $\mathbf{x}_u$ and $\mathbf{x}_i$ are the one-hot encodings of IDs of user $u \in \mathcal{U}$ and item $i \in \mathcal{I}$ with $\mathcal{U}$ and $\mathcal{I}$ being the set of users and items, respectively.

C. Neighbor Aggregation

Neighbor aggregation layers aim to capture CF signals along graph structures and enrich the basic embeddings of users and items, $\mathbf{e}_u^{(0)}, \mathbf{e}_i^{(0)} \in \mathbb{R}^d$, by aggregating information from their neighbors in the graphs.

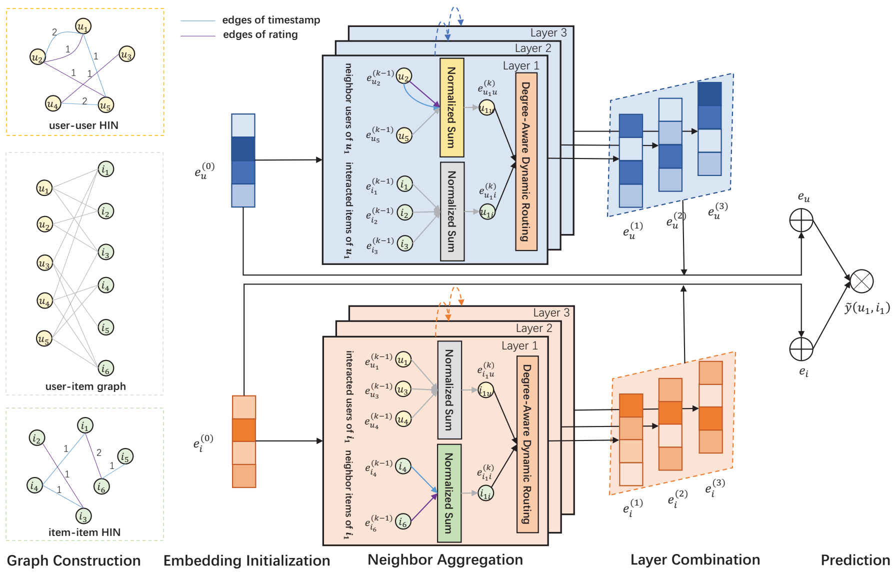


Fig. 2. The overall architecture of our MNR-GCF framework.

1) Multi-Relation Aggregation: Recall that the heterogeneous information networks (HINs) are constructed with multiple types of edges, where each type corresponds to a specific interaction attribute $a \in \mathcal{A}$, and reflects one of the potential neighbor relations. To better distinguish the influences of different potential neighbor relations, we propose a multi-relation aggregation mechanism to aggregate information from neighbors in HINs via *relation-aware self gating mechanism*, which determines what information of neighbors can be aggregated to the next layer according to the relation:

$$\begin{aligned}\mathbf{e}_{u\leftarrow v}^{(k)} &= \sum_{a\in\mathcal{A}}\sum_{v\in\mathcal{M}_u^a}\tilde{\mathbf{A}}_{uv}^{\mathsf{U},a}(\mathbf{e}_v^{(k-1)}\otimes\sigma(\mathbf{W}_a\mathbf{e}_v^{(k-1)}+\mathbf{b}_a)),\\ \mathbf{e}_{i\leftarrow j}^{(k)} &= \sum_{a\in\mathcal{A}}\sum_{j\in\mathcal{M}_i^a}\tilde{\mathbf{A}}_{ij}^{\mathsf{I},a}(\mathbf{e}_j^{(k-1)}\otimes\sigma(\mathbf{W}_a\mathbf{e}_j^{(k-1)}+\mathbf{b}_a)).\end{aligned} \tag{4}$$

Here $\mathbf{e}_{u\leftarrow v}^{(k)}$ and $\mathbf{e}_{i\leftarrow j}^{(k)}$ are the embeddings of user u and item i after aggregating information from their neighbors under multiple relations. $\mathcal{M}_*^a$ is the set of immediate neighbors of a user or an item under attribute a. $\mathbf{W}_a \in \mathbb{R}^{d\times d}$ and $\mathbf{b}_a \in \mathbb{R}^d$ are the attribute-specific, i.e., relation-specific parameters, which are shared between users and items. $\sigma(\cdot)$ is the sigmoid activation function, and $\otimes$ is the element-wise product. $\mathbf{e}_v^{(k-1)}$ and $\mathbf{e}_j^{(k-1)}$ are the final embeddings of user v and item j after fusing aggregated information from direct user-item interactions and multiple neighbor relations at layer $k-1$, which will be explained in detail in the next subsection. $\tilde{\mathbf{A}}_{uv}^{\mathsf{U},a}$ and $\tilde{\mathbf{A}}_{ij}^{\mathsf{I},a}$ are the corresponding elements in the Laplacian normalized adjacency matrix of HINs under attribute a, which can be formulated as:

$$\tilde{\mathbf{A}}_{uv}^{\mathsf{U},a} = \frac{\mathbf{A}_{uv}^{\mathsf{U},a}}{\sqrt{D_u^{\mathsf{ho},a}}\sqrt{D_v^{\mathsf{ho},a}}},\quad \tilde{\mathbf{A}}_{ij}^{\mathsf{I},a} = \frac{\mathbf{A}_{ij}^{\mathsf{I},a}}{\sqrt{D_i^{\mathsf{ho},a}}\sqrt{D_j^{\mathsf{ho},a}}}, \tag{5}$$

where $D_*^{\mathsf{ho},a}$ is the degree of a user or an item in the HINs under attribute a, e.g., $D_u^{\mathsf{ho},a} = \sum_{v\in\mathcal{M}_u^a}\mathbf{A}_{uv}^{\mathsf{U},a}$.

As for the user-item bipartite graph, the aggregation rules are identical to those of LightGCN [8]:

$$\begin{aligned}\mathbf{e}_{u\leftarrow i}^{(k)} &= \sum_{i\in\mathcal{N}_u}\frac{1}{\sqrt{D_u^{\mathsf{he}}}\sqrt{D_i^{\mathsf{he}}}}\mathbf{e}_i^{(k-1)},\\ \mathbf{e}_{i\leftarrow u}^{(k)} &= \sum_{u\in\mathcal{N}_i}\frac{1}{\sqrt{D_i^{\mathsf{he}}}\sqrt{D_u^{\mathsf{he}}}}\mathbf{e}_u^{(k-1)},\end{aligned} \tag{6}$$

where $\mathbf{e}_{u\leftarrow i}^{(k)}$ and $\mathbf{e}_{i\leftarrow u}^{(k)}$ are the embeddings after aggregation in the user-item bipartite graph and $\mathcal{N}_*$ is the set of interacted neighbors of a user or an item. D_*^{he} is the degree of a user or an item in the user-item graph, e.g., $D_u^{\mathsf{he}} = |\mathcal{N}_u|$.

2) Degree-Aware Dynamic Routing: Instead of aggregating information from direct user-item interactions and multiple neighbor relations separately and fusing the final outputs for recommendation, we fuse both aggregated information *at each aggregation layer* to better characterize users and items. Mathematically, the fusion operation is defined as

$$\begin{aligned}\mathbf{e}_u^{(k)} &= w_{u\leftarrow i}^{(k)}\cdot\mathbf{e}_{u\leftarrow i}^{(k)} + w_{u\leftarrow v}^{(k)}\cdot\mathbf{e}_{u\leftarrow v}^{(k)},\\ \mathbf{e}_i^{(k)} &= w_{i\leftarrow u}^{(k)}\cdot\mathbf{e}_{i\leftarrow u}^{(k)} + w_{i\leftarrow j}^{(k)}\cdot\mathbf{e}_{i\leftarrow j}^{(k)}.\end{aligned} \tag{7}$$

Here $w_{u\leftarrow i}^{(k)}$ and $w_{i\leftarrow u}^{(k)}$ are the importance weights of aggregated information from direct user-item interactions, and $w_{u\leftarrow v}^{(k)}$ and $w_{i\leftarrow j}^{(k)}$ are the importance weights of aggregated information from multiple neighbor relations at layer k. Since the above aggregated information from different sources represents very different aspects of a user or an item, the weights should be properly initialized and need to be carefully factored in. Inspired by the dynamic routing mechanism's capability of achieving promising results in disentangling users' multiple interests [31], [34], we propose a novel degree-aware dynamic routing mechanism to *dynamically* and *adaptively* balance and fuse the aggregated information, with initialized weights proportional to the node degree. For users, we first initialize the weights via node degrees in different graphs:

$$w^{(1)}_{u\leftarrow i} = \frac{D^{\mathsf{he}}_u}{D^{\mathsf{he}}_u + D^{\mathsf{ho}}_u}, \quad w^{(1)}_{u\leftarrow v} = \frac{D^{\mathsf{ho}}_u}{D^{\mathsf{he}}_u + D^{\mathsf{ho}}_u}, \tag{8}$$

where $D^{\mathsf{ho}}_u = \sum_{a\in\mathcal{A}} D^{\mathsf{ho},a}_u$. Then we update the weights iteratively by considering the affinity between the fused embedding $\mathbf{e}^{(k-1)}_u$ and the aggregated information, $\mathbf{e}^{(k-1)}_{u\leftarrow i}$ and $\mathbf{e}^{(k-1)}_{u\leftarrow v}$:

$$w^{(k)}_{u\leftarrow i} = \frac{\tilde{w}^{(k)}_{u\leftarrow i}}{\tilde{w}^{(k)}_{u\leftarrow i} + \tilde{w}^{(k)}_{u\leftarrow v}}, \; w^{(k)}_{u\leftarrow v} = \frac{\tilde{w}^{(k)}_{u\leftarrow v}}{\tilde{w}^{(k)}_{u\leftarrow i} + \tilde{w}^{(k)}_{u\leftarrow v}}, \tag{9}$$

where $\tilde{w}^{(k)}_{u\leftarrow i} = w^{(k-1)}_{u\leftarrow i} + \mathbf{e}^{(k-1)^\top}_u \tanh\left(\mathbf{e}^{(k-1)}_{u\leftarrow i}\right)$ and $\tilde{w}^{(k)}_{u\leftarrow v} = w^{(k-1)}_{u\leftarrow v} + \mathbf{e}^{(k-1)^\top}_u \tanh\left(\mathbf{e}^{(k-1)}_{u\leftarrow v}\right)$ with $\tanh(\cdot)$ being a nonlinear activation function to increase the model's representation ability. For items, we can initialize and update their weights in a similar way.

D. Layer Combination and Prediction

After a total of K layers of neighbor aggregations and information fusion, we can obtain user u's multiple embeddings $\{\mathbf{e}^{(1)}_u, \mathbf{e}^{(2)}_u, \cdots, \mathbf{e}^{(K)}_u\}$ and item i's embeddings $\{\mathbf{e}^{(1)}_i, \mathbf{e}^{(2)}_i, \cdots, \mathbf{e}^{(K)}_i\}$ from different layers. We combine the initial embedding and the embeddings obtained at each layer to form the final representation of a user or an item:

$$\begin{aligned} \mathbf{e}_u &= \frac{1}{K+1}(\mathbf{e}^{(0)}_u + \cdots + \mathbf{e}^{(K)}_u), \\ \mathbf{e}_i &= \frac{1}{K+1}(\mathbf{e}^{(0)}_i + \cdots + \mathbf{e}^{(K)}_i). \end{aligned} \tag{10}$$

Furthermore, we can obtain user u's or item i's multiple embeddings after aggregating in the user-item bipartite graph (i.e., $\{\mathbf{e}^{(1)}_{u\leftarrow i}, \cdots, \mathbf{e}^{(K)}_{u\leftarrow i}\}$) and the heterogeneous information networks (e.g., $\{\mathbf{e}^{(1)}_{u\leftarrow v}, \cdots, \mathbf{e}^{(K)}_{u\leftarrow v}\}$), which provide complementary information. We can use a method similar to Eq. (10) to get the final representation of a user or an item using only information from direct user-item interactions or multiple neighbor relations. We denote them by $\mathbf{e}_{ui}$, $\mathbf{e}_{uv}$, $\mathbf{e}_{iu}$ and $\mathbf{e}_{ij}$.

The model prediction is done by calculating the inner product of user and item representations:

$$\hat{y}_{ui} = \mathbf{e}^{\mathsf{T}}_u \mathbf{e}_i + \alpha \cdot \mathbf{e}^{\mathsf{T}}_{ui}\mathbf{e}_{iu} + \beta \cdot \mathbf{e}^{\mathsf{T}}_{uv}\mathbf{e}_{ij}, \tag{11}$$

where α and β are trade-off parameters for the direct interactions' ranking score (i.e., $\mathbf{e}^{\mathsf{T}}_{ui}\mathbf{e}_{iu}$) and the multiple relations' ranking score (i.e., $\mathbf{e}^{\mathsf{T}}_{uv}\mathbf{e}_{ij}$), respectively.

Another possible way of utilizing $\mathbf{e}_{ui}$, $\mathbf{e}_{uv}$, $\mathbf{e}_{iu}$ and $\mathbf{e}_{ij}$ is to devise a multi-task learning scheme to leverage additional supervision over the direct interactions' ranking scores and the multiple relations' ranking scores. However, such an attempt affects our model's training stability and leads to worse performance.

TABLE I
THE STATISTICS OF THE DATASETS USED IN THE EXPERIMENTS.

Dataset	#Users	#Items	#Interactions	Density
Gowalla	29, 858	40, 981	1, 027, 370	0.084%
Amazon-Beauty	6, 000	6, 024	47, 831	0.117%
Amazon-Sports	8, 732	9, 128	67, 703	0.085%

E. Model Training

We employ the pairwise Bayesian personalized ranking (BPR) loss [18] to optimize the model parameters $\Theta = \{\mathbf{e}^{(0)}_u, \mathbf{e}^{(0)}_i, \mathbf{W}_a | u \in \mathcal{U}, i \in \mathcal{I}, a \in \mathcal{A}\}$. Specifically, it encourages the prediction scores of a user's historical items to be higher than those of unobserved items:

$$\mathcal{L} = -\sum_{u\in\mathcal{U}} \sum_{i\in\mathcal{N}_u} \sum_{j\notin\mathcal{N}_u} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \|\Theta\|^2_2, \tag{12}$$

where $\sigma(\cdot)$ is the sigmoid activation function, and λ is the coefficient controlling the strength of L_2 regularization. We do not use dropout mechanisms because our solution is so lightweight that enforcing L_2 regularization is sufficient to prevent overfitting.

IV. EXPERIMENTS

In this section, we perform extensive experiments on three real-world datasets to evaluate our proposed MNR-GCF model and answer the following research questions:

- **RQ1:** How does MNR-GCF perform when compared with state-of-the-art CF methods?
- **RQ2:** How do different components in MNR-GCF, especially multiple neighbor relations, contribute to model performance?
- **RQ3:** How do different hyperparameters (e.g., α, β, and K) affect MNR-GCF's performance?

A. Experimental Setup

1) Datasets and Evaluation Metrics: We consider three widely used public benchmark datasets in the experiments: *Gowalla*[1], *Amazon-Beauty*[2] and *Amazon-Sports*[2], which have different properties in terms of domain, size and sparsity. Gowalla is a check-in dataset obtained from the location-based social networking service Gowalla [35], and Amazon-Beauty and Amazon-Sports are representative datasets from the Amazon-Review collection [36]. Table I summarizes the statistics of the three datasets.

For a fair comparison, we follow the settings used in NGCF [7] and LightGCN [8]. For each dataset, the training set is constructed using 80% of the historical interactions of each user, and the remaining is used as the test set. We treat each observed user-item interaction as a positive instance, and adopt the negative sampling strategy to randomly sample unobserved items for users to form negative instances. To ensure the quality of the datasets, we also use the 10-core

[1]https://snap.stanford.edu/data/loc-gowalla.html
[2]https://jmcauley.ucsd.edu/data/amazon/

TABLE II

THE PERFORMANCE COMPARISON WITH BASELINES. ALL IMPROVEMENTS ARE SIGNIFICANT WITH p-VALUE < 0.05 BASED ON PAIRED t-TESTS.

	Gowalla		Amazon-Beauty		Amazon-Sports	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
BPRMF (*UAI'09*)	0.1291	0.1109	0.0231	0.0177	0.0153	0.0140
NeuMF (*WWW'17*)	0.1399	0.1212	0.0249	0.0191	0.0168	0.0152
NGCF (*SIGIR'19*)	0.1569	0.1327	0.0283	0.0267	0.0172	0.0158
LightGCN (*SIGIR'20*)	0.1823	0.1555	0.0507	0.0282	0.0374	0.0219
Multi-GCCF (*ICDM'19*)	0.1610	0.1318	0.0391	0.0273	0.0269	0.0167
NIA-GCN (*SIGIR'20*)	0.1726	0.1358	0.0460	0.0279	0.0315	0.0183
DGCF (*SIGIR'20*)	0.1842	0.1561	0.0443	0.0258	0.0328	0.0169
IMP (*WWW'21*)	<u>0.1869</u>	<u>0.1585</u>	<u>0.0530</u>	<u>0.0294</u>	<u>0.0449</u>	<u>0.0246</u>
UltraGCN (*CIKM'21*)	0.1862	0.1580	0.0487	0.0270	0.0372	0.0203
MNR-GCF (*Ours*)	**0.1930**	**0.1655**	**0.0541**	**0.0301**	**0.0456**	**0.0255**
Improvement	3.26%	4.42%	2.17%	2.38%	1.53%	3.66%
p-value	$8e^{-5}$	$1e^{-4}$	$3e^{-4}$	$1e^{-3}$	$1e^{-2}$	$5e^{-3}$

setting, i.e., retaining users and items with at least ten interactions. The evaluation metrics we consider are Recall@20 and NDCG@20, identical to the metrics used in NGCF and LightGCN.

B. Baseline Algorithms

To demonstrate the effectiveness of our solution, we compare it with a wide range of representative methods, including four direct user-item interaction based methods and five potential neighbor relation enhanced methods.

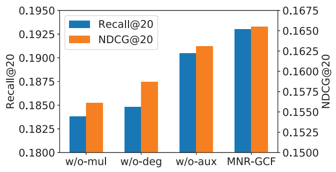
- **BPRMF** [18] is a matrix factorization based method optimized by the BPR loss.
- **NeuMF** [19] is a state-of-the-art neural CF model that uses multiple hidden layers to capture nonlinear relations between users and items.
- **NGCF** [7] is a seminal GCN-based CF model that captures the high-order connectivity between users and items by stacking multiple aggregation layers on user-item bipartite graphs.
- **LightGCN** [8] is a GCN-based CF model evolved from NGCF, which simplifies the design of the feature aggregation component by removing non-linear activations and transformation matrices.
- **Multi-GCCF** [11] employs a multi-graph encoding layer to integrate the information from the user-item, user-user, and item-item graphs.
- **NIA-GCN** [6] is a state-of-the-art GCN-based CF model, which aggregates both neighbors and interacted nodes from a user-item graph via a cross-depth ensemble layer.
- **DGCF** [31] considers user-item interactions at a finer granularity by iteratively refining intent-aware interaction graphs and representations.
- **IMP** [32] decomposes a user-item graph into a set of subgraphs consisting of users with similar interests and performs high-order graph convolution inside the subgraphs.
- **UltraGCN** [12] directly approximates the limit of infinite-layer graph convolutions via constraint losses for efficient recommendations.

1) Implementation Details: Identical to the settings of NGCF and LightGCN, the embedding size is fixed to 64, and the embedding parameters are initialized with the Xavier method [37] for all models. We optimize our model with Adam [38] and use the default learning rate of 0.001 and default mini-batch size of 2048. The L_2 regularization coefficient λ is set to 10^{-4}, and the default number of layers K is set to 3. The trade-off parameters α and β are both searched in the range of $\{0.05, 0.1, 0.2, 0.5\}$ on a validation dataset which is a random 10% subset of the training set and set to 0.2 by default. We implemented our model in PyTorch. The hyperparameters of all baseline algorithms are carefully tuned by grid search. All experiments were run on a workstation with an Intel Xeon Platinum 2.40GHz CPU, an NVIDIA Quadro RTX 8000 GPU and 500GB RAM.

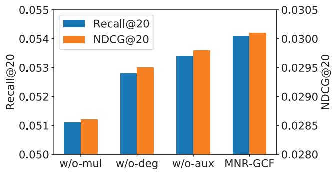
C. Comparison with Baselines (RQ1)

We report the main results in Table II, where the best results are boldfaced and the second-best results are underlined. We can draw a few interesting observations.

- Our solution consistently yields the best performance on three datasets. Its relative improvements over the strongest baselines are 3.26%, 2.17% and 1.53% in terms of Recall@20 and 4.42%, 2.38% and 3.66% in terms of NDCG@20 on Gowalla, Amazon-Beauty and Amazon-Sports, respectively. All improvements are significant with p-value < 0.05 based on paired t-tests. We attribute such improvements to a few reasons. First, explicitly modeling *multiple neighbor relations* among users or items is essential for learning informative user and item representations. Second, by *dynamically* and *adaptively* fusing information from different sources while respecting their different extents of contribution, MNR-GCF can better reflect users' real preferences.
- Compared with the strongest baseline IMP, which aggregates information from multiple decomposed user-item subgraphs, MNR-GCF consistently achieves better performance, validating the necessity of discriminating the information from direct user-item interactions and



(a) Gowalla



(b) Beauty

Fig. 3. Impact of different components on MNR-GCF.

potential neighbor relations and the effectiveness of explicitly constructing heterogeneous information networks to capture multiple potential neighbor relations.

- Compared with another latest baseline UltraGCN, which derives constraint losses on both user-item and item-item relations and adjusts the importance of different relations with predefined hyperparameters, MNR-GCF dynamically and adaptively fuses the aggregated information from two different sources by adjusting their importance at *each aggregation layer* of a GCN, leading to consistently better performance on all three datasets.

D. Impact of Different Components (RQ2)

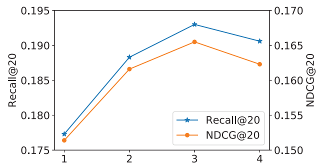
To verify the benefits of different components in MNR-GCF, we conduct an ablation study with several variants:

- **w/o-mul** constructs heterogeneous information networks based on only one type of attributes of user-item interactions, which means only one type of potential neighbor relations is captured.
- **w/o-deg** fuses aggregated information from different sources equally via mean pooling without adopting the degree-aware dynamic routing mechanism.
- **w/o-aux** calculates the final ranking score without considering the auxiliary terms $\mathbf{e}_{ui}$, $\mathbf{e}_{uv}$, $\mathbf{e}_{iu}$ and $\mathbf{e}_{ij}$ in Eq. (11).

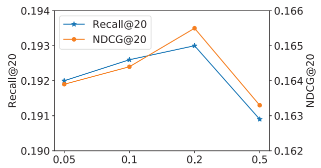
Fig. 3 shows the performance of different variants. We only report the results on Gowalla and Amazon-Beauty, and the observations are similar on the Amazon-Sport dataset. It can be seen that modeling multiple neighbor relations among users or items consistently improves the performance by a significant margin. Adaptively weighing the different importance of aggregated information from direct user-item interactions and multiple neighbor relations and incorporating the auxiliary terms into the ranking score can further improve the performance.

E. Impact of Hyperparameters (RQ3)

We also study the impact of different hyperparameters. We vary the number of neighbor aggregation layers K in the range of $\{1, 2, 3, 4\}$ and report the results in Fig. 4(a). We only report the results on Gowalla. The observations are similar on other datasets. We can observe that initially aggregating more layers can better capture high-order connectivity and lead to substantially better performance. However, stacking too many layers may introduce much noise and suffer from the over-smoothing problem, resulting in worse performance. In addition, since the trade-off parameters in the final ranking score play a pivotal role, we report their impact on the performance in Fig. 4(b). For simplicity, we set α equal to β and search in the range of $\{0.05, 0.1, 0.2, 0.5\}$. We can observe that MNR-GCF achieves the best performance with $\alpha = \beta = 0.2$ on Gowalla, indicating that the auxiliary terms of ranking scores can positively contribute to model performance.



(a) Layer number K



(b) Trade-off parameters α & β

Fig. 4. Impact of different hyperparameters on MNR-GCF.

V. Conclusion

The success of CF in the era of GCNs relies on not only an effective method to precisely characterize multiple relations of users and items in terms of graphs, but also carefully regulated aggregation of information from different relations. In this paper, motivated by the notable limitations of existing GCN-based CF methods, we proposed a novel neural method called MNR-GCF, which features multiple neighbor relations in terms of heterogeneous information networks by considering multiple attributes associated with interactions. A multi-relation aggregation mechanism was presented, which uses a relation-aware self gating mechanism to aggregate information from neighbors in multiple relations. We further introduced a degree-aware dynamic routing mechanism to dynamically and adaptively fuse information from direct user-item interactions and multiple neighbor relations at each aggregation layer, which is distinctive from existing ideas. We conducted a comprehensive experimental evaluation to show that our MNR-GCF framework consistently and significantly outperforms a large number of state-of-the-art competitors on three public benchmark datasets that represent different application domains and that are of different data properties.

References

[1] J. Wang, K. Ding, L. Hong, H. Liu, and J. Caverlee, "Next-item recommendation with sequential hypergraphs," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 1101–1110, 2020.

[2] J. Wang, R. Louca, D. Hu, C. Cellier, J. Caverlee, and L. Hong, "Time to shop for valentine's day: Shopping occasions and sequential recommendation in e-commerce," in *Proceedings of the 13th International Conference on Web Search And Data Mining (WSDM)*, pp. 645–653, 2020.

[3] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proceedings of the 24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1059–1068, 2018.

[4] T. Chen and R. C. Wong, "An efficient and effective framework for session-based social recommendation," in *Proceedings of the 14th International Conference on Web Search And Data Mining (WSDM)*, pp. 400–408, 2021.

[5] M. Liu, J. Li, G. Li, and P. Pan, "Cross domain recommendation via bi-directional transfer graph collaborative filtering networks," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM)*, pp. 885–894, 2020.

[6] J. Sun, Y. Zhang, W. Guo, H. Guo, R. Tang, X. He, C. Ma, and M. Coates, "Neighbor interaction aware graph convolution networks for recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 1289–1298, 2020.

[7] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 165–174, 2019.

[8] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and 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 (SIGIR)*, pp. 639–648, 2020.

[9] E. Min, Y. Rong, T. Xu, Y. Bian, P. Zhao, J. Huang, D. Luo, K. Lin, and S. Ananiadou, "Masked transformer for neighhourhood-aware click-through rate prediction," *CoRR*, vol. abs/2201.13311, 2022.

[10] Z. Li, Y. Zheng, S. Wu, X. Zhang, and L. Wang, "Heterogeneous graph collaborative filtering," *CoRR*, vol. abs/2011.06807, 2020.

[11] J. Sun, Y. Zhang, C. Ma, M. Coates, H. Guo, R. Tang, and X. He, "Multi-graph convolution collaborative filtering," in *Proceedings of the 19th IEEE International Conference on Data Mining (ICDM)*, pp. 1306–1311, 2019.

[12] K. Mao, J. Zhu, X. Xiao, B. Lu, Z. Wang, and X. He, "Ultragcn: Ultra simplification of graph convolutional networks for recommendation," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM)*, pp. 1253–1262, 2021.

[13] J. Yu, H. Yin, J. Li, Q. Wang, N. Q. V. Hung, and X. Zhang, "Self-supervised multi-channel hypergraph convolutional network for social recommendation," in *Proceedings of the Web Conference 2021*, pp. 413–424, 2021.

[14] S. Ji, Y. Feng, R. Ji, X. Zhao, W. Tang, and Y. Gao, "Dual channel hypergraph collaborative filtering," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 2020–2029, 2020.

[15] S. Liu, I. Ounis, C. Macdonald, and Z. Meng, "A heterogeneous graph neural model for cold-start recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 2029–2032, 2020.

[16] C. Chen, W. Ma, M. Zhang, Z. Wang, X. He, C. Wang, Y. Liu, and S. Ma, "Graph heterogeneous multi-relational recommendation," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 3958–3966, 2021.

[17] Y. Koren, R. Bell, and C. Volinsky, "Matrix factoriza-tion techniques for recommender systems.," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[18] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 452–461, 2009.

[19] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web (WWW)*, pp. 173–182, 2017.

[20] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 335–344, 2017.

[21] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong, "Deep item-based collaborative filtering for top-n recommendation," *ACM Transactions on Information Systems*, vol. 37, no. 3, pp. 1–25, 2019.

[22] R. van den Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," in *KDD Workshop on Deep Learning Day*, 2017.

[23] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and 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 (KDD)*, pp. 974–983, 2018.

[24] J. Ma, P. Cui, K. Kuang, X. Wang, and W. Zhu, "Disentangled graph convolutional networks," in *Proceedings of the 36th International Conference on Machine Learning ICML*, vol. 97, pp. 4212–4221, 2019.

[25] L. Chen, L. Wu, R. Hong, K. Zhang, and M. Wang, "Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 27–34, 2020.

[26] H. Zhang and J. J. McAuley, "Stacked mixed-order graph convolutional networks for collaborative filtering," in *Proceedings of the 2020 SIAM International Conference on Data Mining SDM*, pp. 73–81, 2020.

[27] T. Kong, T. Kim, J. Jeon, J. Choi, Y. Lee, N. Park, and S. Kim, "Linear, or non-linear, that is the question!," in *Proceedings of the 15th International Conference on Web Search And Data Mining (WSDM)*, pp. 517–525, 2022.

[28] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 6861–6871, 2019.

[29] J. Song, C. Chang, F. Sun, X. Song, and P. Jiang, "Ngat4rec: Neighbor-aware graph attention network for recommendation," *CoRR*, vol. abs/2010.12256, 2020.

[30] Y. Yang, L. Wu, R. Hong, K. Zhang, and M. Wang, "Enhanced graph learning for collaborative filtering via mutual information maximization," in *Proceedings of the 44rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 71–80, 2021.

[31] X. Wang, H. Jin, A. Zhang, X. He, T. Xu, and T.-S. Chua, "Disentangled graph collaborative filtering," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 1001–1010, 2020.

[32] F. Liu, Z. Cheng, L. Zhu, Z. Gao, and L. Nie, "Interest-aware message-passing GCN for recommendation," in *Proceedings of the Web Conference 2021*, pp. 1296–1305, 2021.

[33] J. Li, Y. Wang, and J. J. McAuley, "Time interval aware self-attention for sequential recommendation," in *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM)*, pp. 322–330, 2020.

[34] C. Li, Z. Liu, M. Wu, Y. Xu, H. Zhao, P. Huang, G. Kang, Q. Chen, W. Li, and D. L. Lee, "Multi-interest network with dynamic routing for recommendation at tmall," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management(CIKM)*, pp. 2615–2623, ACM, 2019.

[35] D. Liang, L. Charlin, J. McInerney, and D. M. Blei, "Modeling user exposure in recommendation," in *Proceedings of the 25th International Conference on World Wide Web (WWW)*, pp. 951–961, 2016.

[36] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *Proceedings of the 25th International Conference on World Wide Web (WWW)*, pp. 507–517, 2016.

[37] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 249–256, 2010.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.