



# NAH: neighbor-aware attention-based heterogeneous relation network model in E-commerce recommendation

Nan Jiang<sup>1</sup> · Zihao Hu<sup>1</sup> · Jie Wen<sup>2</sup> · Jiahui Zhao<sup>1</sup> · Weihao Gu<sup>1</sup> · Ziang Tu<sup>1</sup> · Ximeng Liu<sup>3</sup> · Yuanyuan Li<sup>4</sup> · Jianfei Gong<sup>5</sup> · Fengtao Lin<sup>6</sup>

Received: 21 July 2022 / Revised: 7 November 2022 / Accepted: 1 February 2023 /  
Published online: 10 March 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Traditional recommender systems only utilize a single user-item interaction behavior as the optimization target behavior. However, multi-behavior recommender systems leverage multiple user behaviors as auxiliary behaviors (favorite and page view), which is more practical. Therefore, recommender systems by exploring patterns of multiple behaviors are of great significance in improving performance. Many previous works toward multi-behavior recommendation fail to capture user preference intensity for different items in the heterogeneous graph. Meanwhile, they also ignore high-order relationships that incorporate user different preference intensity into user-item heterogeneous interactions. To solve the above challenges, we propose a novel multi-behavior recommendation model named neighbor-aware attention-based heterogeneous relation network model in E-commerce recommendation (NAH). NAH leverages the attention propagation layer to capture user preference intensity for different items and employs a composition method to incorporate relation embeddings into node embeddings for high-order propagation. Experiment results on two real-world datasets verify the effectiveness of our model in the multi-behavior task by comparing it with some start-of-the-art methods. Further studies verify that our model has a significant effect on exploring high-order information and cold-start users who have few user-item interaction records.

**Keywords** E-commerce recommendation · Graph neural networks · Multi-behavior task · Embedding learning

## 1 Introduction

Recommender systems have been widely used in various fields and can effectively alleviate the issue of information overload. Traditional recommender systems usually only utilize a

---

This article belongs to the Topical Collection: *Special Issue on Privacy and Security in Machine Learning*  
Guest Editors: Jin Li, Francesco Palmieri and Changyu Dong

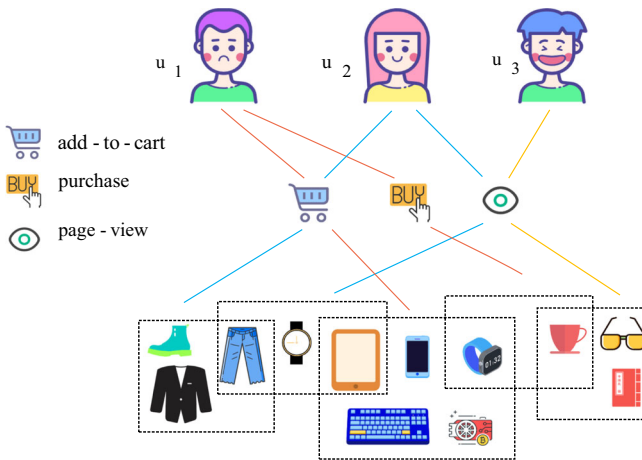
✉ Nan Jiang  
jiangnan@ecjtu.edu.cn

Extended author information available on the last page of the article.

single type of user-item interaction data [1] (i.e., e-commerce, purchase behavior using user-item), but it has data sparsity and cold-start issues. Therefore, more and more works have begun to consider the incorporation of multiple auxiliary behaviors to enhance user preferences, and the approach using multiple behaviors is actually more practical. Figure 1 shows an example of multi-behavior recommendation in E-commerce platform. With the increase in the variety and quantity of items, the data provided by a single behavior (purchase) is limited, and the introduction of auxiliary behaviors such as page-view, favorite, and add-to-cart to infer user preferences will attract much attention.

In recent years, how to take advantage of auxiliary behaviors has become a trending topic, and many works have also proposed many novel methods and achieved excellent performance. Summarizing existing methods [2, 3] for multi-behavior or multi-relation recommendation, Neural Multi-Task Recommendation(NMTR) [4] proposed to build a model using a multi-task learning framework and shared embedding layers, and using Neural collaborative filtering [1] as the score prediction function. Efficient Heterogeneous Collaborative Filtering (EHCF) [5] believed that there is a transitive relationship between various behaviors, and designed a new non-sampling transfer learning scheme to improve the recommendation performance. Multiplex graph neural networks (MGNN) [6] used multiple network structures and graph convolutional networks to learn shared embedding and behavior-specific embedding for nodes. Graph heterogeneous multi-relational recommendation (GHCF) [7] revealed the uncovering relationships between heterogeneous user-item interactions and embedded both embeddings of nodes and relations to exploit the high-order information in heterogeneous graph. Despite effectiveness, they cannot consider the preference strength of neighbor nodes and the high-order relationship under the multi behavior message passing architecture of different nodes.

Motivation by the above observations in previous multi-behavior recommendation works, there are two major issues in Graph Neural Networks (GNNs). The first issue is that GNNs' propagation weights are based on conventional aggregation methods, in which propagation weight in most methods depends on the neighboring nodes or the set of neighboring nodes. However, this does not take into account the intensity of the target node's preference for its neighbors. For instance, Figure 1 shows an example of behavior heterogeneity



**Figure 1** An example of the heterogeneity in E-commerce scenario. Specifically, multiple implicit feedback provide richer information than single target behavior

by incorporating multiple feedback to enhance recommendation. User  $u_1$  purchase smart watch and mug in Figure 1. As we all know, a smart watch provides more user preferences' information than a mug, and  $u_1$  purchase a smart watch and a mug but if their neighboring nodes sets are a similar size. For the user node, the weight of links between them are similar in size. In fact, the link to the smart watch should gain more weight than the link to the mug, so we should take into account user preference intensity for neighboring nodes. In short, multi-behavior recommendation is required to consider the preference strength from neighboring nodes rather than obtaining node weights based on conventional aggregation methods

The second issue of GNNs is that traditional recommender system does not consider high-order relationships between nodes(users and items) by incorporating multiplex behaviors under message passing architecture. The effectiveness of previous multi-behavior methods relies on sufficient user-item interactions to learn better embeddings. We believe that exploring user-item high-order relationships can better capture the connectivity between users and items under sparse relational data. As shown in Figure 1, there are few purchase behavior records. Traditional recommendation methods have intensively solved the relationship between target node and neighboring nodes to learn better embedding. Therefore, we try to alleviate the cold start problem by considering the introduction of auxiliary behaviors and the high-order relations between user-item heterogeneous interaction

To be more specific, we build a unified heterogeneous graph containing two types of nodes (users and items) and edges of different user-item interaction under different behaviors. Firstly, in order to capture node preference intensity, we proposed to leverage the attention mechanism to distinguish different neighboring nodes and assign different weights according to the importance of different items during embedding propagation, and recursively propagate the embedding from the neighboring nodes to update embeddings. Secondly, by considering the behavior relationships between user-item heterogeneous interaction, we incorporate the relationship between users and items into the heterogeneous graph together to utilize the high-hop signals between nodes(users and items) to build a unified multi-behavior prediction model. By incorporating user preference intensity into user-item interaction in the heterogeneous graph, we leverage collaborative signals from high-order neighbors to learn better node embeddings in graph neural networks for enhanced recommendation.

The main contributions of this paper are as follows:

Considering capturing users' preference intensity for different items, the traditional rule-based method of calculating the propagation weight has defects. We exploit the attention embedding propagation layer [8], which propagates the embedding from the neighbors of the node to update its embedding in a recursive way. During the learning propagation weight, weight is calculated according to the importance of different items, which can obtain different contributions from neighboring nodes.

Considering the high-order relationship propagation between user and item under multi-behavior passing architecture, the previous works lacked explicit modeling of the user-item heterogeneous interaction in the high-hop graph structure. We utilize relation-aware propagation layers, which incorporate relation embedding into nodes embeddings to high-order propagation with the hip-hop graph structure of the user-item heterogeneous interaction. And it can fully utilize node information by exploring the high-hop heterogeneous connection, which is helpful to capture high-order relations in nodes embedding learning.

We conduct extensive experiments on two real-world datasets Beibei and Taobao. And the experimental results show that NAH model effectively improves the recommendation

performance compared with the state-of-the-art baselines, which is extremely useful for recommendation tasks to cold-start users.

## 2 Problem scope

In this paper, we aim to further study introducing multiple auxiliary behaviors data in heterogeneous graphs to alleviate the problems of data sparsity and cold start in recommender systems that rely on massive amounts of implicit feedback data. Since implicit feedback lacks user explicit preference information, we focus on how to extract users' different preference intensity to provide more practical research for multi-behavior recommendation tasks.

Therefore, our main research is to incorporate multiple auxiliary behaviors into the recommender system and combine the weights of user-adaptive learning neighbors to perform high-order propagation to learn better node representations. On the one hand, we aim to employ graph neural network's special designs for heterogeneous interaction data to make full use of extracting user preference intensity in recommender systems. On the other hand, we consider the fusion of entities nodes and behaviors for propagation to explore their high-order propagation relationships.

Based on the research on multi-behavior recommender systems, there are two challenges existing in our problem: 1) how to capture and calculate user preference intensity and 2) how to perform high-order propagation to obtain semantic information by incorporating user preference intensity from different hops. Motivated by the above challenges, we propose a novel method NAH that leverages the attention propagation layer to capture user preference intensity and employs the composition method to incorporate relation embeddings into node embeddings to high-order propagation in graph neural networks for multi-behavior recommendation.

## 3 Preliminaries

In the e-commerce recommendation scenario, user utilizes different types of behaviors to help recommender systems to achieve the purpose of improving the recommendation performance. According to the introduction, GNNs [9–11] have performed well in recommendation methods due to the powerful learning capability of graph-structured data. single-behavior recommendation [1, 12, 13] may not perform as well as multi-behavior recommendation. To sum up, multi-behavior recommendation is studied from two categories: the first category is to utilize multi-behavior data to improve the sampling strategy of positive and negative samples or to optimize the loss function. In previous work, multi-channel Bayesian personalized ranking (MC-BPR) [14] employs an extended sampling method to obtain different types of feedback reflecting different strengths of user preference according to different types of implicit feedback. Specifically, EHCF [5] designed an efficient heterogeneous collaborative filtering model to capture the interaction behavior between users and items, establish fine-grained user-item relationships, and effectively learn model parameters from pure data to further improve performance. Another category is to leverage multi-behavior data to improve the capability to learn better user and item embeddings. Graph neural networks for social recommendation (GraphRec) [15] provides an approach to jointly capturing interactions and opinions in the user-item graph, which coherently models heterogeneous advantages. Intra- and inter-heterogeneity recommendation (ARGO) [16] explored

a graph-based message-passing architecture to model interaction heterogeneity with relational aggregation networks and recursively propagate the embedding of adjacent nodes on the user-item graph. In our work, we intend to design a recommendation model by leverage multiple type behavior, which takes into account the user preference intensity for the different neighboring items and explores high-order information between nodes(users and items).

---

$p_u^{(0)}, q_v^{(0)}$	initial embeddings for users, items
$U, V$	collection of user sets and collection of item sets
$S$	relation embedding matrix
$\mathcal{N}_u, \mathcal{N}_v$	neighborhood of user and neighborhood of item
$u, v$	a user node and a item node
$M, N$	the number of users and the number of items
$K$	the number of behavior types
$P, Q$	user embedding matrix and item embedding matrix
$s_r^{(0)}$	initial embeddings for relation
$R^{(k)}$	user-item interaction under k-th type behavior
$R^{(k)}_{uv}$	if u has interacted with v under the behavior k
$d$	user and item embedding dimension
$d'$	relation embedding dimension
$W^{(l)}$	the attention network weight matrix
$W_r^{(l)}$	the relation weight matrix
$W_{nn}^{(l)}$	graph neural network weight matrix
$\hat{R}^{(k)}_{uv}$	the predicted probability that u will interact with v under the target behavior

---

Assume that there are two types of entities  $U$  and  $V$ , which denote the set of users and items, respectively. Assume that  $M$  and  $N$  denote the number of users and items respectively, where  $u$  denotes a user, and  $v$  denotes an item. The multiple types behavior of user interaction matrix denote as  $\{R^{(1)}, R^{(2)}, \dots, R^{(K)}\}$ , where  $K$  denotes the number of behaviors,  $\{R^{(1)}, R^{(2)}, \dots, R^{(K-1)}\}$  denote the auxiliary behaviors, and  $R^{(K)}$  denotes the target behavior.  $R^{(k)}$  denotes whether the user has interacted with the item under behavior k. We suppose that the item of interaction matrix  $R^{(k)}_{uv}$  has a value of 1 or 0:

$$R^{(k)}_{uv} = \begin{cases} 1, & \text{if u has interacted with v under k;} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

In the multi-behavior recommender system, the R-th behavior is generally selected as the target behavior to be optimized. The target behavior is usually purchase behavior in the e-commerce scenario. And the auxiliary behavior includes click, favorite, share, add to cart,

etc. Given the target user  $u$ , the multi-behavior recommendation task can be formulated as:

**Input:** user-item interaction data based on multiple types of behaviors  $\{R_1, R_2, \dots, R_K\}$ .

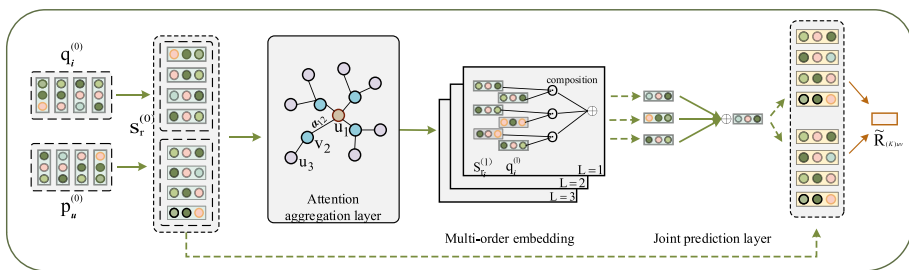
**Output:** Under the target behavior, according to the estimated probability  $\hat{R}_{(K)uv}$ , the items that have not interacted with user  $u$  are ranked top-N and recommended.

### 4 Methodology

In this section, the details of our proposed NAH model will be described, and its architecture is shown in Figure 2. Our model has three important components, 1)a sharing embedding layer that generates initial features for the user, item and behaviors embeddings; 2)a message aggregation layer that aggregates feature information from adjacent vertices to learn user preference intensity and extracts user-item interaction information to high-hop propagation based on multi-behavior data; 3) joint prediction layers to fuse each layer embedding, which will be used to predict the likelihood that the user will interact with items under target behavior.

#### 4.1 Unified heterogeneous graph

According to the above, we know that the multi-behavior recommendation task is to utilize various auxiliary behaviors to recommend the target user under the target behavior. Therefore, we aim to build a unified heterogeneous graph to model the research problem. For an undirected graph  $G = (V, E, R)$ , node set  $V$  contains user node  $u \in U$  and item node  $v \in V$ . The edges in  $E$  denote edges of different user-item interactions under different behaviors, and relation  $R$  represents the set of all behavior types. Specifically, user  $u_1$  adds item  $v_1$  to cart under auxiliary behavior  $r_2$ , then there exists an edge in graph  $G$ , denoted as  $R_{u_1, v_1}^{r_2} = 1$ . In general, graph  $G$  is used to propagate and update node embedding, during which the neighborhood messages are aggregated based on behavior-aware interaction information between users and items. As far as our method is concerned, we assign different weight for neighboring node embedding propagation that user-item interaction depends on user preference so that justifies the necessity of capturing user preference intensity in multi-behavior task. In order to make better use of multi-behavioral



**Figure 2** The illustration of NAH model architecture. Given user  $u$ , item  $i$  and relation  $S_r(r)$  as the input. 1.Attention Aggregation Layers: aggregate the neighboring node messages and relational embedding; 2.Multi-order embedding: aggregate high-order relations by performing the composition operation on adjacent node  $v$  under its relation  $r$ . 3.Joint prediction layer: get user, item and  $u$ - $i$  interaction embedding behavior after propagating  $L$  layers

data, the user-item propagation architecture explores high-hop connectivity that incorporates user preference intensity to capture more accurate behavioral collaborative signals for multi-behavior recommendation.

## 4.2 Sharing embedding layers

Similar to the existing multi-behavior methods [4, 7, 17], we employ one-hot encoding to the input user and item IDs. Let  $p_u^{(0)} \in \mathcal{R}^d$  and  $q_v^{(0)} \in \mathcal{R}^d$  represent the features of user  $u$  and item  $v$  respectively, where  $d$  is embedding dimension. Let  $\mathbf{P} = \{p_{u_1}^{(0)}, p_{u_2}^{(0)}, \dots, p_{u_M}^{(0)}\}$  and  $\mathbf{Q} = \{q_{v_1}^{(0)}, q_{v_2}^{(0)}, \dots, q_{v_N}^{(0)}\}$  represent the embedding matrix of user and item respectively, and ID embedding layer can be defined as a fully connected layer.

$$p_u^{(0)} = \mathbf{P} \cdot X_u^T, q_v^{(0)} = \mathbf{Q} \cdot Y_v^T \quad (2)$$

where the size of  $\mathbf{P}$  and  $\mathbf{Q}$  are  $M \times d$  and  $N \times d$ , and represent the  $u$ -th and  $v$ -th row vectors in  $\mathbf{P}$  and  $\mathbf{Q}$ , respectively.  $X_u^T$  and  $Y_v^T$  denote the one-hot feature vectors for user  $u$  and item  $v$ . Note that embeddings matrices  $\mathbf{P}$  and  $\mathbf{Q}$  as initial features of node users and node items can be considered as input features for each node in our framework. For different types of behavior, the initial feature vector of behavior denotes as follows:

$$s_r^{(0)} = \mathbf{S} \cdot Z_r^T \quad (3)$$

where the size of  $\mathbf{S}$  is  $K \times d'$ , and  $K$  represents the number of behavior types,  $d'$  represents relations embedding dimension. Behavior embedding is also generated by an ID embedding layer, which is used to project the behaviors to the user-item vector space.

## 4.3 Message aggregation layers

After receiving embedding merged by a sharing embedding layer, the next step is to aggregate the neighboring node messages and update target node embedding under different type behaviors. In our framework, we aggregate the neighboring nodes to capture user preference intensity by an embedding aggregation mechanism and update user and item embeddings by high-order propagation based on type-aware behavior for recommendation.

### 4.3.1 Embedding aggregation

Our main idea is to consider the user preference intensity on item differently by combining the two key factors of relational embedding and aggregation of messages from the user's neighboring nodes according to different behavior types under the multi-behavior messaging architecture. For each target node, the information of neighboring nodes is fused into the embedding by propagation, which reinforces embedding learning, by considering that the contribution of each item to user preference is different. In our task, to make better use of the neighboring node information, we leverage the attention mechanism to capture the importance of the target node and neighboring nodes, and according to the importance of neighbors, to help learn structure information.

In our message passing architecture, we leverage the attention mechanism to calculate the importance weights of user node  $u$  and neighboring node  $v$ . In particular, we obtain the embedding representation of the corresponding neighbors of the neighboring node  $v$  of node  $u$  through a layer of neural network. Then, we employ the similarity function to calculate

the similarity of the node itself to its neighbors, which is formally calculated as follows:

$$\alpha'_{u_i v_j} = a(\mathbf{W}_1 p_{u_i}, \mathbf{W}_2 q_{v_j}) \quad (4)$$

where  $u_i$  denotes the target user, and  $v_j$  denotes neighboring nodes of  $u_i$ .  $a(\cdot)$  is a similarity function, which represents the similarity between  $u_i$  and  $v_j$ . In this paper, we define  $a(\cdot)$  as a layer neural network.  $\alpha'_{u_i v_j}$  represents the importance of item  $v_j$  to user  $u_i$ . In addition,  $\mathbf{W}_1$  and  $\mathbf{W}_2$  denote trainable transformation matrices.

In order to obtain the attention coefficient easier to calculate and compare, we incorporate the softmax function to regularize all the adjacent nodes  $v_j$  of the target node  $u_i$ . We leverage the broadcast mechanism to get the attention matrix and normalize the target node by a softmax function to calculate the importance weights as follows:

$$\alpha_{u_i v_j} = \frac{\exp(\alpha'_{u_i v_j})}{\sum_{v_j \in \mathcal{N}_u} \exp(\alpha'_{u_i v_j})} \quad (5)$$

where  $\alpha'_{u_i v_j}$  denotes the intermediate value of the input softmax function to generate importance weights  $\alpha_{u_i v_j}$ . Therefore, the embedding aggregation layers formula is as follows:

$$\alpha_{u_i v_j} = \frac{\exp(\sigma(a[\mathbf{W}_1 p_u || \mathbf{W}_2 q_v]))}{\sum_{k \in \mathcal{N}_u} \exp(\sigma(a[\mathbf{W}_1 p_u || \mathbf{W}_2 q_k]))} \quad (6)$$

where  $||$  denotes the concatenation operation and we define  $\sigma(\cdot)$  as Leaky\_ReLU [18] non-linear activation function. And we obtain the importance weights  $\alpha_{u_i v_j}$  between user  $i$  and item  $j$  under behavior type  $k$ . Then, the embedding aggregation process is transformed as follows:

$$p_u^{(l)} = \sigma \left( \sum_{v_j \in \mathcal{N}_u} \alpha_{u_i v_j} \mathbf{W}^{(l-1)} q_v^{(l-1)} \right) \quad (7)$$

where  $\sigma(\cdot)$  is Leaky\_ReLU.  $p_u$  and  $q_v$  are the input user and item embeddings for neural network layer, respectively. Likewise, item embedding also can be based on the above aggregation and propagation process. In our embedding generated process, different embeddings are aggregated by weight  $\alpha_{u_i, v_j}$  and transformation parameter  $\mathbf{W} \in d \times d$  from different latent dimensions. Under behavior type  $k$ , our message passing architecture for the target user node  $u_i$  from its adjacent item nodes proceeds in a similar way in (7).

### 4.3.2 High-order propagation

After performing the aggregation embeddings of type-specific behavior between nodes(users and items), we model the high-order relations by performing the operations on adjacent node  $v$  under its relation  $r$ . Given the generated user-item interaction graph structure, we learn high-order relations in multi-behavior framework on a graph  $G$  by stacking multiple layers of information propagation. According to the weight of the user's behavior, we follow to employ weighted sum as combination.

In high-order propagation process, the embeddings of node users(or node items) is represented by accumulating incoming messages from all heterogeneous interaction items(users).



Inspired by entity-relation composition operations used in knowledge graph embedding approaches[19], The message passing equation of our model is defined as:

$$\begin{aligned}
 q_v^{(l)} &= \sigma \left( \sum_{(u,r) \in \mathcal{N}_v} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_v|}} \mathbf{W}_{nn}^{(l)} \left( p_u^{(l-1)} \odot s_r^{(l-1)} \right) \right) \\
 p_u^{(l)} &= \sigma \left( \sum_{(v,r) \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_v|}} \mathbf{W}_{nn}^{(l)} \left( q_v^{(l-1)} \odot s_r^{(l-1)} \right) \right)
 \end{aligned}
 \tag{8}$$

where  $\mathcal{N}_u$  and  $\mathcal{N}_v$  denote the set of neighbors of users and items, respectively;  $\mathbf{W}_{nn}^{(l)}$  is the graph neural network parameters of the model;  $\sigma$  is Leaky\_ReLU activation function.  $\frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_v|}}$  is the symmetric normalization term, which is used to avoid the increase of the embeddings scale with graph convolution operations while increasing.  $\odot$  denotes the element-wise product of vectors and  $q_v^{(l-1)} \odot s_r^{(l-1)}$  is to incorporate relation embeddings into the message-passing formulation.

After the node embeddings defined in (8) are updated, the relation embeddings are also performed as follows:

$$s_r^{(l)} = \mathbf{W}_r^{(l)} s_r^{(l-1)} \tag{9}$$

where  $\mathbf{W}_r^{(l)}$  is a relational neural network weight parameter that projects all relations as nodes into the same embedding scale and make them available on the next layer.

#### 4.4 Joint prediction layer

In the joint prediction layer, we get a multi-layer representation  $\{p_u^{(0)}, \dots, p_u^{(l)}\}$  for user,  $\{q_v^{(0)}, \dots, q_v^{(l)}\}$  for item,  $\{s_r^{(0)}, \dots, s_r^{(l)}\}$  for user and item interaction embedding behavior after propagating L layers. Obtaining embeddings from different layers indicates that the information of neighbors of different orders is to be a combination. So we further combine them to get the final representation: To predict the likelihood of multiple behaviors of a user towards an item, the learned representation under each behavior type is used as a separate prediction layer.

$$\begin{aligned}
 p_u &= \sum_{k=0}^l \alpha_k p_u^k, \\
 q_v &= \sum_{k=0}^l \alpha_k q_v^k, \\
 s_r &= \sum_{k=0}^l \alpha_k s_r^k
 \end{aligned}
 \tag{10}$$

where  $\alpha_k$  is a hyper-parameter that represents the importance of the k-th layer embedding. Inspired by layer combination approaches to get final representations in simplifying and powering Graph Convolution Network [12], we select uniform weight  $\frac{1}{L+1}$  as a combination operation. By this way, we predict the final embedding with information from different layers, which not only enriches semantic information but also captures the effect of graph convolution with self-connections.

In order to predict the likelihood of multiple user behaviors towards an item, the learned representation for each behavior is incorporated into a separated prediction layer. To be

specific,  $s_r^k$  denotes the learned embedding of the k-th behavior, and user  $u$  estimates the probability of item  $v$  under the k-th behavior as:

$$\hat{R}_{(r)uv} = p_u \cdot \text{diag}(s_r^k) \cdot q_v = \sum_i^d p_{ui} s_{rki} q_{vi} \tag{11}$$

where  $\text{diag}(s_r^k)$  represents the diagonal matrix whose diagonal elements correspond to  $s_r^k$ , and  $d$  represents the embedding size.

### 4.5 Multi-task learning

To learn the parameters, our main idea is that the massive amount of heterogeneous implicit feedback data and data sparsity, the sampling-based learning method will result in a limited number of observed samples with interactions, while large-scale samples without observed interactions are not observed in the recommendation task. Therefore, we decided to adopt the latest high-efficiency Non-negative sampling learning method [5] to optimize our model. In order to better learn the model parameters, we introduce a weighted regression with squared loss [20] to compute the loss for a single behavioral matrix:

$$\mathcal{L}_r = \sum_{u \in U} \sum_{v \in V_u^+ \cup V_u^-} \lambda_{uv}^r (R_{(r)uv} - \hat{R}_{(r)uv})^2 \tag{12}$$

where  $\lambda_{uv}^r$  is the weight of  $R_{(r)uv}$ . Then,  $V_u^+$ ,  $V_u^-$  represent the set of positive and negative items for target user  $u$ , respectively.

Suppose  $\mathcal{L}_r^P$  is the loss of positive data and  $\mathcal{L}_r^A$  is the loss of all data. Based on the latest high-efficiency Non-negative sampling learning method [5],  $\mathcal{L}_r$  is the sum of the loss of positive data and the loss of all data, and the loss of unlabeled data has been eliminated. Thus, the loss for a single behavioral matrix is performed as:  $\mathcal{L}_r = \mathcal{L}_r^P + \mathcal{L}_r^A$ , where

$$\begin{aligned} \mathcal{L}_r^P &= \sum_{u \in U} \sum_{v \in V_u^+} ((\lambda_{uv}^{r+} - \lambda_{uv}^{r-}) \hat{R}_{(r)uv}^2 - 2\lambda_{uv}^{r+} \hat{R}_{(r)uv}) \\ \mathcal{L}_r^A &= \sum_i^d \sum_j^d ((s_{ki} s_{kj}) \left( \sum_{u \in U} p_{ui} p_{uj} \right) \left( \sum_{v \in V} \lambda_{uv}^{r-} q_{vi} q_{vj} \right)) \end{aligned} \tag{13}$$

Finally, following the Multi-Task Learning (MTL) mode, different but related tasks models are jointly trained, and the minimum loss function is performed as:

$$\mathcal{L} = \sum_{r=1}^R \gamma_r \mathcal{L}_r + \mu \|\theta\|_2^2 \tag{14}$$

where  $\gamma_r$  is a hyper-parameter that controls the effect of the r-th behavior on joint training, which is set differently according to different datasets;  $r$  is the number of behavior types. We also carry out that  $\sum_{r=1}^n \gamma_r = 1$  to advance the tuning of hyper-parameters.

To optimize the objective function, we use mini-batch Adam [21] as the optimizer, which enables the learning rate to be self-adaptively updated during training, alleviating the

difficulty of choosing an appropriate learning rate. In our model, we also employ the dropout methods, which is an efficient workaround to prevent overfitting [22] of neural networks.

## 5 Experiments

In this section, we conduct experiments on two real-world datasets in the processing of multi-behavior recommendation tasks to validate the effectiveness of our proposed NAH model by comparing it with the state-of-the-art baselines. We describe the details of the distribution of the data sets, ablation study of auxiliary behavior, and setting of hyper-parameters.

### 5.1 Experimental settings

#### 5.1.1 Datasets

We evaluate the performance of the model on two real-world datasets, collected on **Taobao** and **Beibei** platforms. The specific information of datasets are displayed in Table 1:

**Beibei:** The Beibei dataset [4] is the largest e-commerce platform for maternal and infant specialty products in China. This dataset includes different types of user-item interaction data, and we take three behavior (page-view, add-to-cart, and purchase) to study their influence on recommendation performance in our experiments.

**Taobao:** The Taobao dataset [23] is the most popular online e-commerce platform in China. Compared with the beibei dataset, they have the same three user behaviors, but the data distribution of the two is completely different. Beibei has fewer users and items entities but has many records of auxiliary behaviors and more records of target behaviors. And Taobao has more users and item entities, but fewer behavior records.

For a fair comparison with the start-of-the-art baselines, the datasets were preprocessed consistently following previous works [5], which split both datasets according to the number of records of the target behavior and exclude users and items with less than 5 target behaviors.

#### 5.1.2 Evaluation metrics

To evaluate our model performance, we employ widely used leave-one-out techniques [4, 5] and employ Hit Rate (HR) [26] and Normalized Discounted Cumulative Gain (NDCG) [27]. HR is a common indicator to measure recall rate, which represents the more items are on the top-N list. NDCG is an evaluation metric for ranking results, which emphasizes the impact of the item's position in the top-N list. For users, our evaluation scheme is to rank all unlabeled items in the training set, so we get more convincing results than randomly

**Table 1** Statistical details of the evaluation datasets

Dataset	#User	#Item	#View	#Add-to-cart	#Purchase
Beibei	21,716	7,977	2,412,586	642,622	304,576
Taobao	48,749	39,493	1,548,126	193,747	259,747

sampled subset rankings. By doing so, the results are more convincing than only ranking a random subset of negative items [28, 29].

### 5.1.3 Baselines

To verify the effectiveness of our method, we selected several state-of-the-art models to compare with our model. And we classify them into two categories based on one-behavior models and multi-behavior models.

one-behavior model:

**BPR** [24]: it is a widely used learning framework that optimizes pairwise loss for item recommendation.

**NCF** [1]: a state-of-the-art learning framework by introducing neural network to learn the user-item interaction information for item ranking.

**LightGCN** [12]: a simplified GCN model that only includes the most important parts and improves the recommendation performance.  
multi-behavior model:

**CMF** [25]: it is a popular technique that factorizes multiple matrices jointly to boost the overall factorization quality by decomposing the rating matrix  $R$ , side information matrix of user and item, respectively.

**MC-BPR** [14]: the adaptive negative sampling rule in BPR and uses level information to sample negative samples to expand BPR for heterogeneous data.

**NMTR** [4]: it uses a cascaded way to build user behavior relationships and uses multi-task learning to model from users' multi-behavior data.

**EHCF** [5]: it applied an efficient non-sampling strategy (Non-Sampling, Whole-data based Learning) to a multi-behavior recommender system for the first time, and achieves very significant results in training time and model performance.

**GHCF** [7]: this is a state-of-the-art method that reveals latent relationships between heterogeneous user-item interactions and exploits a based on relationship-aware GCN propagation layer to acquire high-hop signals.

### 5.1.4 Parameters settings

For parameters, we initialize the latent factor dimension with 64-dim and set the batch size to 256. And we further explore optimal parameters on validation dataset and evaluate the model on test dataset. For the parameters of baseline, we mainly set the parameters according to the parameters and tuning strategies provided by the original model. In training, we optimize our model as mini-batch Adagrad [30] optimizer. Then, we set the learning rate as 0.001 and utilize the Xavier initialization [31] to initialize the parameters. Besides, we set the message dropout ratio  $\rho$  as 0.2 and the node dropout ratio as 0.1 to prevent overfitting in our model. Regularization coefficient is set to 10 for Beibei and 0.01 for Taobao. Furthermore, We utilize early stop to avoid overfitting, where the training process will be stopped if recall@10 on validation set does not increase within 50 epochs. For other hyper-parameters, uniform negative entry weight is set to 0.1 for Beibei and 0.01 for Taobao and multi-task learning coefficient  $\gamma$  is set to  $\gamma_1 = 1/6$ ,  $\gamma_2 = 4/6$ ,  $\gamma_3 = 1/6$  for sampling-based methods in the baseline. For the non-sampling methods in the baseline, the negative sampling ratio is set is 4, the negative weight values for Beibei and Taobao are set to 0.01 and 0.1 respectively, and experiments show good performance.

## 5.2 Overall performance

In this subsection, we selected several state-of-the-art baselines to compare with our model. Then, the performance of all models on two datasets is shown in Tables 2 and 3. Note that we refer to the parameters and tuning strategies provided by the original models to make a fair comparison. In order to investigate the Top-N performance, we set length N to [10, 50, 100, 200] in our experiments. From the experiment results, the followings can be observed:

**Model effectiveness** From the figure, we can find that on HR and NDCG evaluation metrics, our NAH substantially outperforms state-of-the-art baselines on both datasets. For instance, the average improvement of our model over the state-of-the-art baseline is 2.96% and 1.51% on the Beibei dataset and 2.87% and 1.46% on Taobao dataset for Recall and NDCG, which clarifies the performance of our model. Our model also demonstrates that the effectiveness of information of the neighbors of nodes are incorporated into their embeddings.

**User preference intensity** Among other single-behavior baselines, models based on graph neural network (i.e., LightGCN) achieve better performance in most cases. This illustrates the rationality of collaborative signals encoding based on user-item graph relationship for embedding propagation between adjacent nodes. However, LightGCN and NMTR fail to learn better the target node representation from neighboring nodes, so it demonstrates that is very important to capture user preference intensity for neighboring nodes for recommendation performance, which enhances the ability of representation learning.

**Mutil-behavior model** In comparisons with BPR, NCF and LightGCN, we can observe that incorporating multi-behavior information into predicting can generally outperform methods using only a single behavior, which illustrates the importance of heterogeneous data for recommendation performance. For instance, The improvement of our model over the state-of-the-art single-behavior method is 98.5% and 71.5% on Beibei and Taobao dataset for HR@100, respectively. This illustrates that incorporating multiple types of behaviors to the embedding function of recommender systems plays a positive role.

**Non-sampling learning strategy** the non-sampling methods(EHCF,GHCF,ARGO,NAH) generally outperform methods that sampling-based methods (NCF,LightGCN,MC-BPR,NMTR). EHCF demonstrates the effectiveness of using the whole-data-based learning strategy, which is suitable for learning from heterogeneous behavior data.

## 6 Discussions

In this section, we discuss the impact of auxiliary behavior, impact of data sparsity, hyperparameter sensitivity and possible limitations of our model.

### 6.1 Impact of auxiliary behaviors

To understand the effectiveness of multiple auxiliary behaviors, we explore the impact of auxiliary behaviors on the model performance on both datasets with the consideration of different type behavior from multiple behaviors.

NAH-P: the model variant of only purchase.

**Table 2** NAH recommendation performance on Beibei comparing with start-of-the-art baselines

Beibei	Method	HR@10	HR@50	HR@100	HR@200	NDCG@10	NDCG@50	NDCG@100	NDCG@200
Single-behavior	<b>BPR</b> [24]	0.0431	0.1247	0.2183	0.3056	0.0211	0.0415	0.0541	0.0681
	NCF [1]	0.0437	0.1565	0.2348	0.3573	0.0226	0.0450	0.0591	0.0759
	LightGCN [12]	0.0448	0.1618	0.2484	0.3730	0.0231	0.0463	0.0613	0.0795
Heterogeneous-behavior	<b>CMF</b> [25]	0.0480	0.1585	0.2856	0.4311	0.0245	0.0459	0.0663	0.0843
	MC-BPR [14]	0.0501	0.1734	0.2723	0.3849	0.0250	0.0504	0.0657	0.0794
	NMTR [4]	0.0523	0.2057	0.3180	0.4741	0.0288	0.0610	0.0765	0.0973
	EHCF [5]	0.1507	0.3305	0.4300	0.5445	0.0815	0.1204	0.1364	0.1530
	GHCF [7]	0.1900	0.3712	0.4617	0.5563	<b>0.1008</b>	0.1414	0.1557	0.1690
	NAH	<b>0.1922</b>	<b>0.3878</b>	<b>0.4773</b>	<b>0.5752</b>	0.1006	<b>0.1443</b>	<b>0.1588</b>	<b>0.1726</b>

**Table 3** NAH recommendation performance on Taobao comparing with start-of-the-art baselines

Taobao	Method	HR@10	HR@50	HR@100	HR@200	NDCG@10	NDCG@50	NDCG@100	NDCG@200
Single-behavior	<b>BPR</b> [24]	0.0373	0.0708	0.0878	0.1034	0.0225	0.0271	0.0308	0.0331
	NCF [1]	0.0387	0.0723	0.0890	0.1056	0.0231	0.0276	0.0315	0.0341
	LightGCN [12]	0.0409	0.0806	0.1018	0.1203	0.0234	0.0327	0.0367	0.0383
Heterogeneous-behavior	<b>CMF</b> [25]	0.0479	0.0776	0.1174	0.1558	0.0253	0.0297	0.0351	0.0376
	MC-BPR [14]	0.0541	0.0786	0.1272	0.1598	0.0262	0.0289	0.0357	0.0392
	NMTR [4]	0.0581	0.0946	0.1367	0.1859	0.0278	0.0339	0.0395	0.0537
	EHCF [5]	0.0709	0.1601	0.2198	0.2911	0.0404	0.0588	0.0684	0.0792
	GHCF [7]	0.0805	0.1844	0.2506	0.3301	<b>0.0448</b>	0.0673	0.0781	0.0890
	NAH	<b>0.0815</b>	<b>0.1905</b>	<b>0.2598</b>	<b>0.3441</b>	0.0444	<b>0.0681</b>	<b>0.0805</b>	<b>0.0913</b>

NAH-PV: the model variant of including purchase and page view.

NAH-PC: the model variant of including purchase and add to cart.

Tables 4 and 5 show the model performance for different combination behaviors. From the results, auxiliary behaviors both page view data and add to cart data can lead to better recommendation performance. And the performance of NAH is further improved when all three behavior data are used simultaneously. This demonstrates the effectiveness of modeling auxiliary behaviors for user preference. Besides, we find that two observations: On one hand, carting behavior on Beibei dataset has a greater effect on the recommendation than carting behavior on Taobao dataset, which may be due to the amount of auxiliary behavior data on two datasets. On the other hand, we incorporate auxiliary behaviors into our model, which significantly improves the recommendation performance. It can be seen that the impact of user implicit feedback on recommender systems, which also makes the research on multi-behavior recommendation meaningful.

## 6.2 Impact of data sparsity

Data sparsity is a big challenge since there are lacking of record of target behavior and for new users without behavior record, which make most recommendation models inefficient. However, we enhance the recommendation by introducing auxiliary behavior [32] and  $u$ - $i$  high-order relationship. From Tables 2 and 3, the average improvement of our model is superior to the state-of-the-art baseline based on optimal parameters. The improvement of our model over the state-of-the-art single-behavior method is 139.7% and multi-behavior method is 4.47% on Beibei dataset for HR@50, illustrating the strong power of NAH model. From the Figure 3, our NAH outperforms other methods, including the state-of-the-art multi-behavior models such as NMTR and GHCF. In particular, our method performed well for users with less than 16 purchase records, which verifies the validation of alleviating data sparsity issues. Moreover, the performance on Taobao dataset and Beibei dataset is slightly different, we think this may be caused by the different data distribution of the dataset. For example, Taobao dataset is sparser than Beibei dataset and has fewer user-item interactions; page view data is much more than add-to-cart data due to page view data are easier to collect; Taobao dataset and Beibei dataset have great differences in the number of purchase behavior records in each data sparsity split. The results demonstrate the effectiveness of NAH in alleviating the data sparsity issue with auxiliary behaviors since NAH learns target behavior and auxiliary behaviors in a reasonable way.

## 6.3 Hyper-parameter study

To evaluate how hyper-parameters affect the performance of our NAH, we investigate the effect of two important hyper-parameters coefficient  $\gamma_k$  and layer numbers  $L$  in our model. Since our model is a multi-task model, we test three loss coefficients  $\gamma_1$ ,  $\gamma_2$ , and  $\gamma_3$  in the multi-task loss function. In addition, we analyze the influence of the layer numbers  $L$  on performance.

Firstly, we examine the influence of different  $\gamma_k$  on Beibei and Taobao datasets to check. We tune the three loss coefficients in  $[0, 1/6, 2/6, 3/6, 4/6, 5/6, 1]$ . As  $\gamma_1 + \gamma_2 + \gamma_3 = 1$ , when  $\gamma_1$  and  $\gamma_2$  are given, the value of  $\gamma_3$  is determined. When  $\gamma_1=0$  and  $\gamma_2=0$ , the model only has purchase behavior like single behavior recommendation, and it performs badly. For both datasets, setting  $(1/6, 4/6, 1/6)$  achieves the best performances.

Next, we test the influence of the depth of our model on Beibei and Taobao datasets to check the effectiveness of multiple embedding propagation layers. In particular, we test



**Table 4** Performance of variants of NAH model on Beibei

Beibei	behavior	HR@10	HR@50	HR@100	HR@200	NDCG@10	NDCG@50	NDCG@100	NDCG@200
Ablation	NAH-P	0.0516	0.1613	0.2438	0.3476	0.0251	0.0482	0.0613	0.0756
	NAH-PV	0.0739	0.2402	0.3539	0.4808	0.0343	0.0694	0.0876	0.1051
	NAH-PC	0.1886	0.3795	0.4700	0.5624	0.0967	0.1392	0.1539	0.1668
Model	<b>NAH</b>	<b>0.1922</b>	<b>0.3878</b>	<b>0.4773</b>	<b>0.5752</b>	<b>0.1006</b>	<b>0.1443</b>	<b>0.1588</b>	<b>0.1726</b>

**Table 5** Performance of variants of NAH model on Taobao

Taobao	behavior	HR@10	HR@50	HR@100	HR@200	NDCG@10	NDCG@50	NDCG@100	NDCG@200
Ablation	<b>NAH-P</b>	0.0367	0.0755	0.0966	0.1209	0.0204	0.0289	0.0323	0.0357
	NAH-PV	0.0618	0.1673	0.2390	0.3262	0.0318	0.0545	0.0661	0.0783
	NAH-PC	0.0607	0.1639	0.2336	0.3235	0.0315	0.0537	0.0649	0.0775
Model	<b>NAH</b>	<b>0.0815</b>	<b>0.1905</b>	<b>0.2598</b>	<b>0.3408</b>	<b>0.0444</b>	<b>0.0680</b>	<b>0.0805</b>	<b>0.0913</b>

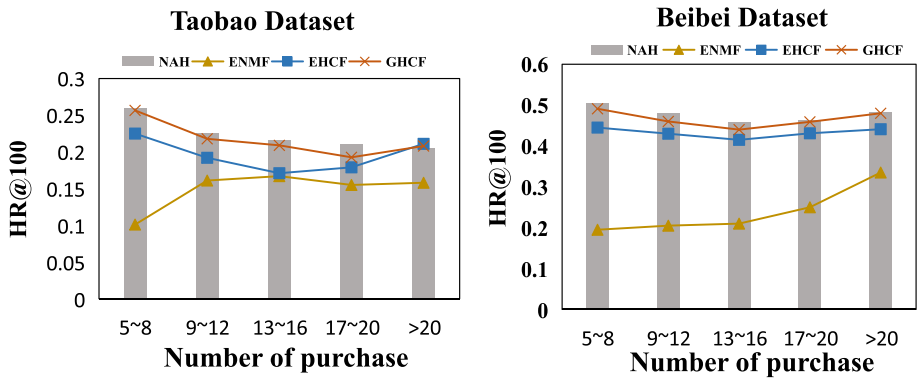


Figure 3 Impact of number of Purchase

vary the depth of our model in the range of [1,2,3,4,5] and the results are shown in Figure 4. When L=1, it represents that the model has a first-order embedding propagation layer and others are similar. In the figure, the y-axis denotes the performance of HR@100 compared with different layer numbers. From the figure, we can observe that by increasing the model depth from 1 to 4 on the Beibei and Taobao datasets, the performance of our model is significantly improved, which demonstrates that our model has the capability to capture high-order relationships in multi-behavior recommendation scenario.

### 7 Conclusion

In this paper, we investigate the issue of multi-behavior recommendation that considers user preference intensity and high-hop propagation based on heterogeneous user feedback. To fully model user preference intensity between users and items under different types of behaviors, We propose a novel graph-based approach NAH. The proposed NAH has leveraged the attention mechanism to obtain the important weight of different neighborhoods and explore high-order propagation on a heterogeneous graph. Extensive experimental results

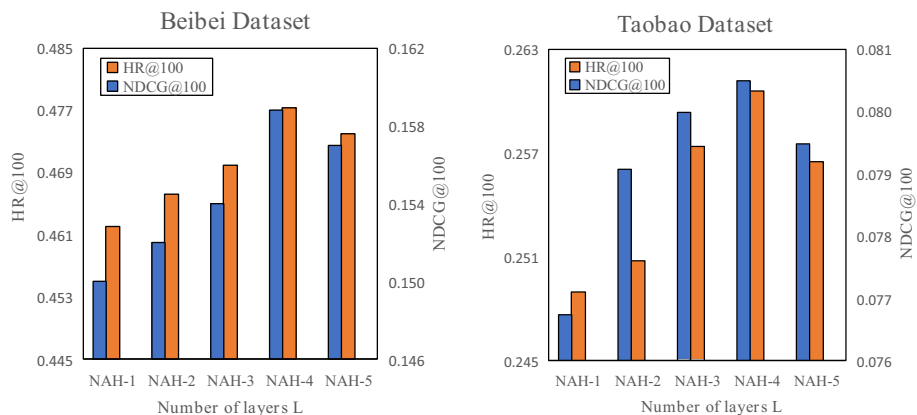


Figure 4 Impact of model depth

demonstrate the state-of-the-art performance of NAH on two real-world datasets. Further ablation studies verify the effectiveness of employing different types of auxiliary behaviors and alleviating data sparsity issues in our NAH.

Although specifically designed for extracting user preference intensity, we explore the high-order relationship under the multi-behavior recommendation, the experiment proves that the depth of embedding propagation layers still has certain limitations. In the future, we will further explore how to solve the problems of overfitting or data noise. We are also interested in exploring special designs on a heterogeneous graph, such as adaptive learning behavior importance and semantics. In addition, we also intend to extend NAH model to other heterogeneous graph recommendations scenarios.

**Acknowledgements** This work is supported by National Natural Science Foundation of China under Grant No. 62172160 and 62062034, Jiangxi Provincial Natural Science Foundation under Grant No. 20212ACB212002, Excellent Scientific and Technological Innovation Teams of Jiangxi Province under Grant No. 20181BCB24009.

**Author Contributions** Among the authors in the list, Nan Jiang took charge of researching and revising it critically for intellectual content. Zihao Hu mainly wrote the manuscript and designed the model. Jie Wen and Yuanyuan Li participated in the design of this study. JiahuiZhao and Weihao Gu collected important background information. Ziang Tu and Ximeng Liu designed of the study and the conception. Jianfei Gong and Fengtao Lin revised the manuscript. After consultations, all the authors agreed with the addition of authors in this paper, and all the authors agreed with their arrangement of the names. In the final version of the article, Nan Jiang is tagged as the corresponding author.

**Funding** The fundings conclude National Natural Science Foundation of China under Grant No. 62172160 and 62062034, Jiangxi Provincial Natural Science Foundation under Grant No. 20212ACB212002, Excellent Scientific and Technological Innovation Teams of Jiangxi Province under Grant No. 20181BCB24009.

**Availability of supporting data** Beibei dataset and Taobao dataset are public datasets available.

## Declarations

**Ethics approval and consent to participate** Not applicable.

**Human and Animal Ethics** Not applicable.

**Consent for Publication** Not applicable.

**Competing interests** We declare that we have no conflict of interest.

## References

1. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.-S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182 (2017)
2. Ding, J., Yu, G., He, X., Quan, Y., Li, Y., Chua, T.-S., Jin, D., Yu, J.: Improving implicit recommender systems with view data. In: IJCAI, pp. 3343–3349 (2018)
3. Guo, L., Hua, L., Jia, R., Zhao, B., Wang, X., Cui, B.: Buying or browsing?: predicting real-time purchasing intent using attention-based deep network with multiple behavior. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1984–1992 (2019)
4. Gao, C., He, X., Gan, D., Chen, X., Feng, F., Li, Y., Chua, T.-S., Jin, D.: Neural multi-task recommendation from multi-behavior data. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE), pp. 1554–1557. <https://doi.org/10.1109/ICDE.2019.00140> (2019)
5. Chen, C., Zhang, M., Zhang, Y., Ma, W., Liu, Y., Ma, S.: Efficient heterogeneous collaborative filtering without negative sampling for recommendation. In: Proceedings of the AAAI conference On Artificial Intelligence, vol. 34, pp. 19–26 (2020)

6. Zhang, W., Mao, J., Yi, C., Xu, C.: Multiplex graph neural networks for multi-behavior recommendation. In: Proceedings of the 29th ACM International Conference On Information & Knowledge Management, pp. 2313–2316 (2020)
7. Chen, C., Ma, W., Zhang, M., Wang, Z., He, X., Wang, C., Liu, Y., Ma, S.: Graph heterogeneous multi-relational recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 3958–3966 (2021)
8. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. *Stat.* **1050**, 20 (2017)
9. Gori, M., Pucci, A., Roma, V., Siena, I.: Itemrank: a random-walk based scoring algorithm for recommender engines. In: IJCAI, vol. 7, pp. 2766–2771 (2007)
10. Yang, J.-H., Chen, C.-M., Wang, C.-J., Tsai, M.-F.: Hop-rec: high-order proximity for implicit recommendation. In: Proceedings of the 12th ACM Conference on Recommender Systems, pp. 140–144 (2018)
11. Page, L., Brin, S., Motwani, R., Winograd, T.: The Pagerank Citation Ranking: bringing order to the Web. Technical report, Stanford InfoLab (1999)
12. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: 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, pp. 639–648. <https://doi.org/10.48550/arXiv.2002.02126> (2020)
13. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.-S.: Kgat: knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 950–958. arXiv:1905.07854. <https://doi.org/10.48550> (2019)
14. Loni, B., Pagano, R., Larson, M., Hanjalic, A.: Bayesian personalized ranking with multi-channel user feedback. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 361–364 (2016)
15. Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., Yin, D.: Graph neural networks for social recommendation. In: The World Wide Web Conference, pp. 417–426 (2019)
16. Xia, L., Huang, C., Xu, Y., Dai, P., Lu, M., Bo, L.: Multi-behavior enhanced recommendation with cross-interaction collaborative relation modeling. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE), pp. 1931–1936. <https://doi.org/10.1109/ICDE51399.2021.00179> (2021)
17. Jin, B., Gao, C., He, X., Jin, D., Li, Y.: Multi-behavior recommendation with graph convolutional networks. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 659–668 (2020)
18. Maas, A.L., Hannun, A.Y., Ng, A.Y., et al.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml, vol. 30, pp. 3. Citeseer (2013)
19. Vashishth, S., Sanyal, S., Nitin, V., Talukdar, P.: Composition-based multi-relational graph convolutional networks. arXiv:1911.03082. <https://doi.org/10.48550> (2019)
20. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: 2008 Eighth IEEE International Conference On Data Mining, pp. 263–272. Ieee (2008)
21. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv:1412.6980. <https://doi.org/10.48550> (2014)
22. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
23. Zhu, H., Li, X., Zhang, P., Li, G., He, J., Li, H., Gai, K.: Learning tree-based deep model for recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1079–1088. arXiv:1801.02294. <https://doi.org/10.48550> (2018)
24. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: bayesian personalized ranking from implicit feedback. arXiv:1205.2618. <https://doi.org/10.48550> (2012)
25. Zhao, Z., Cheng, Z., Hong, L., Chi, E.H.: Improving user topic interest profiles by behavior factorization. In: Proceedings of the 24th International Conference on World Wide Web, pp. 1406–1416 (2015)
26. Järvelin, K., Kekäläinen, J.: Ir evaluation methods for retrieving highly relevant documents. In: ACM SIGIR Forum, vol. 51, pp. 243–250. ACM New York, NY, USA (2017)
27. Karypis, G.: Evaluation of item-based top-n recommendation algorithms. In: Proceedings of the Tenth International Conference on Information and Knowledge Management, pp. 247–254 (2001)
28. Krichene, W., Rendle, S.: On sampled metrics for item recommendation. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1748–1757 (2020)
29. Dacrema, M.F., Cremonesi, P., Jannach, D.: Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In: Proceedings of the 13th ACM Conference on Recommender Systems, pp. 101–109. arXiv:1907.06902. <https://doi.org/10.48550> (2019)

30. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, vol. 12(7) (2011)
31. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256. *JMLR Workshop and Conference Proceedings* (2010)
32. Lu, W., Jiang, N., Di, J., Chen, H., Liu, X.: Learning distinct relationship in package recommendation with graph attention networks. *IEEE Trans. Computat. Social Syst.*, pp. 1–13 (2022)

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## Affiliations

Nan Jiang<sup>1</sup> · Zihao Hu<sup>1</sup> · Jie Wen<sup>2</sup> · Jiahui Zhao<sup>1</sup> · Weihao Gu<sup>1</sup> · Ziang Tu<sup>1</sup> · Ximeng Liu<sup>3</sup> · Yuanyuan Li<sup>4</sup> · Jianfei Gong<sup>5</sup> · Fengtao Lin<sup>6</sup>

Zihao Hu  
1459509701@qq.com

Jie Wen  
wenjie412@126.com

Jiahui Zhao  
707250862@qq.com

Weihao Gu  
920967706@qq.com

Ziang Tu  
11843004@qq.com

Ximeng Liu  
snbnix@gmail.com

Yuanyuan Li  
YY82126@126.com

Jianfei Gong  
1105071063@qq.com

Fengtao Lin  
menggu.lin@163.com

<sup>1</sup> College of Information Engineering, East China JiaoTong University, NanChang, 330013, China

<sup>2</sup> College of Electrical and Automation Engineering, East China JiaoTong University, NanChang, 330013, China

<sup>3</sup> College of Mathematics and Computer Science, Fuzhou University, Fuzhou, 350108, China

<sup>4</sup> Azman Hashim International Business School, University of Technology Malaysia, Johor, 81310, Malaysia

<sup>5</sup> Jiangxi Academy of Social Sciences, NanChang, 330077, China

<sup>6</sup> Key Laboratory of Conveyance and Equipment of Ministry of Education, East China JiaoTong University, NanChang, 330013, China