

## Article

# Hybrid Graph Neural Network Recommendation Based on Multi-Behavior Interaction and Time Sequence Awareness

Mingyu Jia , Fang'ai Liu <sup>\*</sup>, Xinmeng Li and Xuqiang Zhuang

School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China

<sup>\*</sup> Correspondence: lfa@sdsnu.edu.cn

**Abstract:** In recent years, mining user multi-behavior information for prediction has become a hot topic in recommendation systems. Usually, researchers only use graph networks to capture the relationship between multiple types of user-interaction information and target items, while ignoring the order of interactions. This makes multi-behavior information underutilized. In response to the above problem, we propose a new hybrid graph network recommendation model called the User Multi-Behavior Graph Network (UMBGN). The model uses a joint learning mechanism to integrate user-item multi-behavior interaction sequences. We designed a user multi-behavior information-aware layer to focus on the long-term multi-behavior features of users and learn temporally ordered user-item interaction information through BiGRU units and AUGRU units. Furthermore, we also defined the propagation weights between the user-item interaction graph and the item-item relationship graph according to user behavior preferences to capture more valuable dependencies. Extensive experiments on three public datasets, namely MovieLens, Yelp2018, and Online Mall, show that our model outperforms the best baselines by 2.04%, 3.82%, and 3.23%.

**Keywords:** multi-behavior recommendation; sequential recommendation; graph neural network; embedding propagation



**Citation:** Jia, M.; Liu, F.; Li, X.; Zhuang, X. Hybrid Graph Neural Network Recommendation Based on Multi-Behavior Interaction and Time Sequence Awareness. *Electronics* **2023**, *12*, 1223. <https://doi.org/10.3390/electronics12051223>

Academic Editor: Alberto Fernandez Hilario

Received: 25 December 2022

Revised: 27 February 2023

Accepted: 2 March 2023

Published: 3 March 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Recommendation systems have been widely used in various Internet business services in the era of big data. The recommendation model can recommend products that match its users for various businesses and find suitable user groups for enterprises [1]. In order to better personalize recommendations for each user, it is crucial to fully understand the interests and behavioral preferences of users. For sales platforms, understanding users' purchasing interests and behavioral preferences can increase sales and profit margins. For the user him/herself, identifying the user's shopping interests and behavioral preferences on the client side can improve the user's experience and save unnecessary browsing time. The early popular collaborative filtering algorithm (CF) decomposes a single user-item interaction into latent representations for finding similar users and related items and then predicting the next user behavior [2,3]. However, since traditional CF cannot model user attributes and item auxiliary information, there are data-sparsity and cold-start problems in practical application scenarios [4]. To address these issues, supervised learning (SL) models such as Factorization Machine (FM) [5] and NFM (Neural FM) [6] have emerged one after another. With the development of neural network techniques, collaborative filtering architectures for enhancing nonlinear feature interactions utilize multilayer perceptrons to handle advanced nonlinear relationships, such as NCF [7] and DMF [8].

In recent years, deep neural networks based on graph data have received extensive attention, showing good results in processing high-dimensional sparse user interaction data. These neural network structures, called graph neural networks [9,10], are used to learn meaningful representations in graph data structures. Since user-item interactions are often sparse non-Euclidean data, graph data structures can be used to store their

interactions. In addition, the introduction of external Knowledge Graph (KG) data can expand the additional information about users and items [11]. This provides a feasible solution for improving the accuracy and interpretability of recommendation systems. Given the strong performance of graph neural networks in aggregating and propagating graph-structured data, it provides an unprecedented opportunity to improve the performance of recommendation systems.

However, recommendation systems based on graph neural networks also face many problems: (1) Different graph data provide user and item information from different perspectives. How to aggregate and learn more accurate node representations from different types of graphs is crucial for recommendation models [12]. (2) The connections between nodes are diverse rather than single [13]. The assignment of weights to different connection methods requires more consideration. (3) Graph neural networks show good performance in learning the relationships between nodes. However, it is difficult for them to process sequence information [14]. Therefore, it is worth considering how to incorporate temporal information into the model. In this paper, our research question is how to utilize multi-behavior interaction time-series information for an accurate recommendation.

Because of the limitations of existing graph network methods, it is crucial to develop a hybrid graph neural network model that focuses on user behavioral characteristics and user-item interaction habits. Therefore, we designed a user multi-behavior awareness module and an item-information-relation module based on the graph neural network. Specifically, we propose a new method called the User Multi-Behavior Graph Neural Network (UMBGN) Hybrid Model, which has four sections. (1) User-item connection weight calculation: It provides unique weight information for each edge to describe the connection relationship between nodes according to the multi-behavior interaction information between users and items. (2) User-item graph network information transfer: It aggregates the feature information of the node's neighbors according to the edge weights to obtain the final feature representation. (3) Information perception based on user behavior sequence: It uses a behavior-aware network module with bidirectional GRU and AUGRU to enrich the user's behavioral information representation, fully considering the user's behavioral characteristics. (4) Information aggregation between items. It aggregates user-item interaction information by using an attention mechanism and considers the order of interactions between items. Compared with traditional graph network models, our model computes weight information between nodes according to different behavioral interactions. This allows for a more accurate dissemination of information between neighboring nodes. Furthermore, compared with the existing state-of-the-art graph neural network recommendation models, our proposed method introduces user multi-behavior sequential information perception, achieving more accurate recommendation performance. This benefits from the fact that our model considers not only the global nature of multi-behavior interactions but also each user's personality. Therefore, the contributions of the paper can be summarized as follows:

(1) We constructed a user multi-behavior awareness module with bidirectional GRU and AUGRU to enrich user-behavior-information representation. We input the user's interaction with items into the network in chronological order to obtain the user's interaction behavior feature vector, which helps us understand the user's behavioral preferences. Then we integrate the interaction behavior feature vector with the user's feature vector to more accurately locate the user's interest.

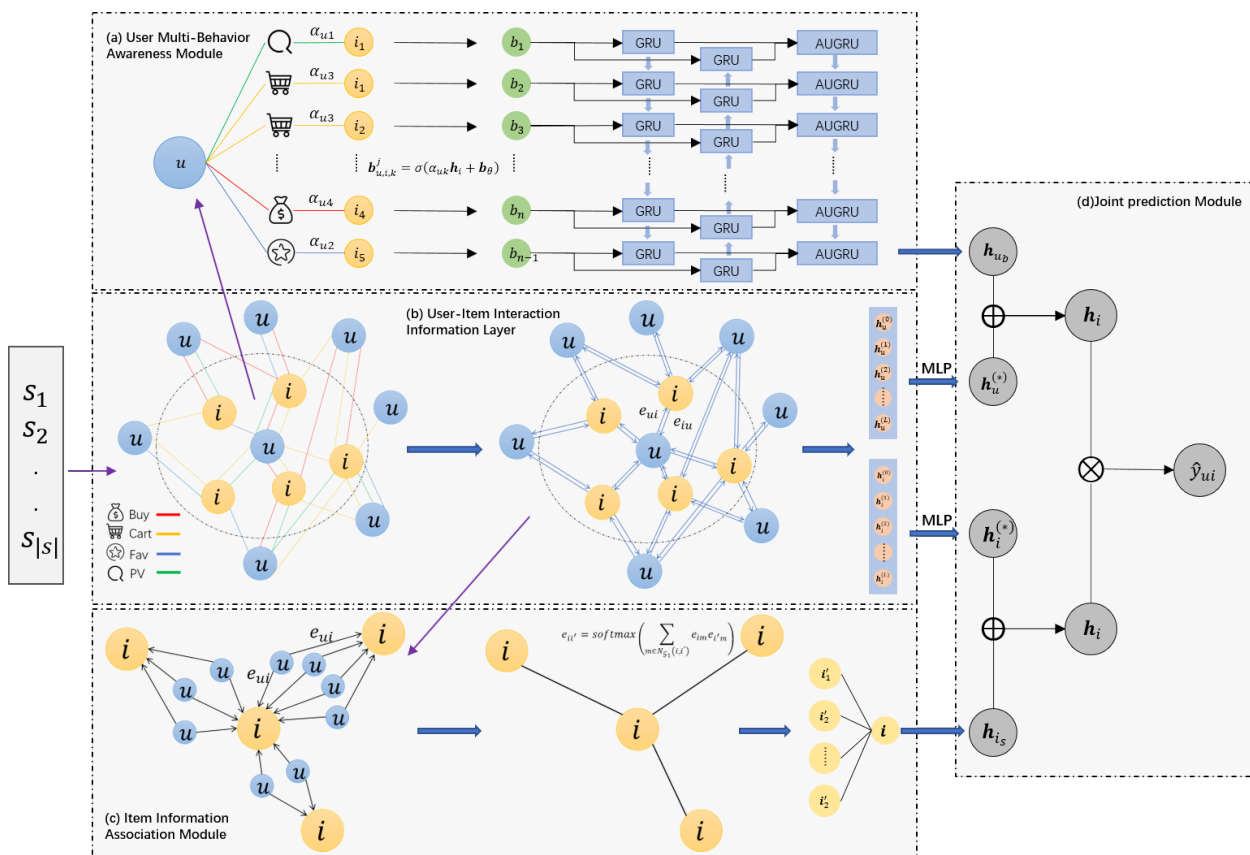
(2) We propose the connection weights between user-item nodes by focusing on user-item multi-behavior interaction information to make information aggregation and dissemination more accurate. In addition, we design an item-information relation module based on the user's dependencies on items. Then we use the attention mechanism to aggregate the item-item connections information to further enrich the embedding representation of items.

(3) The experiments performed on three real datasets indicate that our UMBGN model achieves significant improvements over existing models. In addition, we also

extensively studied the overall impact of different modules on the experiments to prove the effectiveness of our method.

## 2. Methods

In this section, we elaborate on our method; the basic architecture is shown in Figure 1. Our model consists of four modules: (1) a user–item interaction information module, which mines user–item multi-behavior interaction information; (2) a user multi-behavior awareness module, which further learns the strength of each user interaction behavior and extracts long-term user behavior preference; (3) an item-information-relation module, which, according to the user–item interaction information, calculates the information of other items related to an item; and (4) a joint prediction module, which combines the information of each module to obtain the final output result.



**Figure 1.** The framework of the UMBGN model. It contains four modules: module (a) is used to extract user multi-behavior interaction information, module (b) is used to extract user long-term multi-behavior preferences, module (c) is used to extract association information between items, and module (d) is used to output the result.

### 2.1. Symbol Description

We use the set  $U = \{u_1, u_2, \dots, u_m\}$  to represent user information, where  $m$  is the total number of users. Similarly, we use set  $I = \{i_1, i_2, \dots, i_n\}$  to represent item information, where  $n$  is the total number of items. The set  $K = \{k_1, k_2, \dots, k_L\}$  is used to represent user interactions with items (e.g., favorites, purchases, and clicks).

User–item interaction sequence: In the recommendation scenario, we usually obtain the historical sequence of user–item interactions and the time information of their interactions, defined as  $S = \{s_1, s_2, \dots, s_{|S|}\}$ . Moreover,  $s_i = (u, i, k, t)$  indicates that user  $u$  interacts with item  $i$  through behavior  $k$  at time  $t$ .

Input: User–item multi-behavior interaction sequence  $S = \{s_1, s_2, \dots, s_{|S|}\}$ .

Output: The probability,  $\hat{y}_{(p,q)}$ , that user  $u_p$  interacts with item  $i_q$ , with which he/she has no interaction.

## 2.2. Preliminary Preparation

### 2.2.1. Generation of the Bipartite Graph of User–Item Interactions

Our task is to use various interaction information to make recommendations for target users. According to Zhang et al. [15,16], the interaction information between users and items is sparse with non-Euclidean data, and building a knowledge graph can better represent the relationship between them. Therefore, we generate an extended user–item interaction graph  $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0)$ , using the user–item interaction data  $S = \{s_1, s_2, \dots, s_{|S|}\}$ , where node  $\mathcal{V}_0$  consists of user node  $u \in U$  and item node  $i \in I$ . Similar to existing graph models in [17], the  $d$ -dimensional vectors  $p_u$  and  $p_i$  are used to represent the user and item embeddings. The edge, set  $\mathcal{E}_0$ , is a two-tuple composed of interaction type,  $k \in K$ , and timestamp information,  $t$ , denoted as  $\mathcal{E}_0 = (k, t)$ . Different edges represent different behaviors. This can help extract behavior-based information between users and items.

### 2.2.2. User’s Behavior Interaction Information Extraction

Traditional graph network recommendation often only focuses on the users’ single-behavior interaction information, ignoring the influence of edge sets on information dissemination in the graph network. In this paper, we add a certain weight to the edge set of the graph network according to the user–item interaction behavior. This optimizes the process of information transfer in the graph network. We consider two factors that affect user–item interaction preferences: the relative importance of interactions and the temporal order of interactions. On the one hand, users have their own unique interactive behavior habits. For example, user  $u_1$  likes to favorite items, but user  $u_2$  prefers to put the favorite items in the shopping cart. Then their unique behaviors have different relative importance. On the other hand, items also have unique interactions with users. For example, item  $i_1$  is usually favorited by users, but item  $i_2$  is usually added to the shopping cart by users. Therefore, we design different interactive behavior weights,  $\alpha_{uk}$  and  $\alpha_{ik}$ , between users and items:

$$\alpha_{uk} = \frac{w_k^u \cdot n_{uk}}{\sum_{m \in N(u)} w_m^u \cdot n_{um}}; \alpha_{ik} = \frac{w_k^i \cdot n_{ik}}{\sum_{m \in N(i)} w_m^i \cdot n_{im}}, \quad (1)$$

where  $w_k^u$  and  $w_k^i$  are learnable parameters, representing the degree of influence of users and items on behavior,  $k$ ;  $n_{uk}$  represents the number of items that user  $u$  interacts with through type  $k$ ;  $n_{ik}$  represents the number of users that user  $i$  interacts with through type  $k$ ;  $N(u)$  represents all items interacting with user  $u$ ; and  $N(i)$  represents all users interacting with the item  $i$ .

## 2.3. User–Item Multi-Behavior Interaction Information Transfer and Aggregation

### 2.3.1. Construction of User–Item Relationship Graph

We not only pay attention to local relations but also global interaction relations to learn user–item interaction multi-behavior information. According to the user–item interaction graph,  $\mathcal{G}_0$ , we calculate the weight of the edge set through normalization and obtain the connection strength information,  $e_{ui}$  and  $e_{iu}$ , between every two points:

$$e_{ui} = \sigma \left( \sum_{k \in N(u,i)} \alpha_{uk} + b \right); e_{iu} = \sigma \left( \sum_{k \in N(i,u)} \alpha_{ik} + b \right), \quad (2)$$

where  $\sigma$  is the sigmoid function,  $b$  is the bias, and  $N(i, u)$  is the sum of the interaction types between  $i$  and  $u$ . Then  $e_{ui}, e_{iu} \in \mathcal{E}_1$ , and point set  $\mathcal{V}_0$  are combined to obtain the user–item bidirectional relationship graph,  $\mathcal{G}_1 = (\mathcal{V}_0, \mathcal{E}_1)$ . Compared with the traditional

undirected graph network, the bidirectional graph network with weight information has better performance in information transmission.

### 2.3.2. Information Dissemination of User–Item Relationship Graph

It is an effective method to use graph neural networks to analyze graph data structures [9,10]. These networks used an iterative message aggregation method to mine structural information within node neighborhoods. According to the method of Xiang et al. [18], our graph network has a total of  $L$  layers and follows their aggregation and propagation method. Firstly, the nodes in the graph network aggregate the information of their neighbor nodes in the previous layer. Then they update themselves by combining the aggregated information with their original information. Different from Xiang et al., we designed a propagation weight according to the connection strength of nodes in order to achieve a better information transmission effect. Specifically,

$$h_u^{(l)} = \varphi \left( W_1 h_u^{(l-1)} + \sum_{i \in N(u)} \lambda_u^i W_2 h_i^{(l-1)} \right), \quad (3)$$

where  $h_u^{(l)} \in R^d$  is the user's embedding in the  $l$ -th layer,  $h_i^{(l)} \in R^d$  is the item's embedding in the  $l$ -th layer;  $h_u^{(0)} = p_u, h_i^{(0)} = p_i$ ;  $\varphi$  represents the LeakyReLU function for information transformation; and  $W_1$  and  $W_2 \in R^{d \times d}$  are learnable weight matrices. Moreover,  $\lambda_u^i$  is the attention coefficient of user  $u$  to item  $i$ , and its calculation formula is as follows:

$$\lambda_u^i = \frac{\exp(e_{ui})}{\exp\left(\sum_{j \in N(u)} e_{uj}\right)}. \quad (4)$$

Similarly, we can obtain the  $l$ -th embedding information,  $h_i^{(l)}$ , of item node  $i$ . After embedding propagation, neighborhood information is fused into each node's embedding information. To obtain a better representation of the nodes' information, we use a standard multilayer perceptron (*MLP*) to combine the  $L$  layers embedding representations of nodes into the final embedding representation. Among them, all the embedding information of the  $L$  layers is concatenated together before being input into the *MLP*. The specific form is as follows:

$$h_u^{(*)} = MLP\left(h_u^{(0)} \| h_u^{(1)} \dots \| h_u^{(L)}\right); h_i^{(*)} = MLP\left(h_i^{(0)} \| h_i^{(1)} \dots \| h_i^{(L)}\right), \quad (5)$$

where *MLP* is a multilayer perceptron;  $\|$  represents the concatenation operation of vectors; and  $h_u^{(*)}$  and  $h_i^{(*)} \in R^d$  are the final embedding representations of user  $u$  and item  $i$ , respectively.

## 2.4. Perceptron Module Based on User–Item Multi-Behavior Interaction Sequence

### 2.4.1. User Multi-Behavior Feature Extraction

The purpose of this module is to aggregate heterogeneous information generated by multi-behavior patterns between users and their interacting items. Different from the extraction of neighbor information, we also mine user multi-behavior embedding features based on user historical interaction behavior sequences. To obtain the preference information of users interacting with items, we designed a user multi-behavior awareness module. This module extracts the target user,  $u$ , and the neighbor nodes,  $i \in N(u)$ , interacting with it, and it arranges them into  $S_u = \{s_u^1, s_u^2, \dots, s_u^{|S_u|}\}$  according to the time sequence.

According to the embedding information of item nodes and edge nodes, we can obtain the behavior characteristics of user  $u$ :

$$b_{u,i,k}^j = \sigma(\alpha_{uk}h_i + b_\theta), \tag{6}$$

where  $h_v \in R^d$  is the embedding representation information of item  $v$ ,  $\sigma$  is the sigmoid function, and  $b_\theta$  is the bias. By using Formula (6), we can obtain an embedding interaction sequence  $(b^1, b^2, \dots, b^{|N(u)|})$  of user  $u$ .

#### 2.4.2. Bi-GRU-Based Behavior Feature Extraction

To mine the overall features of user-embedded behavioral feature sequences, we use an RNN model to explore their temporal information and generate a single representation to encode their overall semantics. Different from basic RNN units, GRU units can memorize long-term dependencies sequentially [19]. Therefore, in this module, we use GRU to capture the user’s multi-behavior preferences. Guo et al. [20] demonstrate that Bi-LSTM and Bi-GRU can achieve better results in sequential problems than LSTM and GRU. Therefore, we input the embedding interaction sequence  $(b^1, b^2, \dots, b^{|N(u)|})$  into a Bi-GRU network:

$$(h_b^1, h_b^2, \dots, h_b^{|N(u)|}) = Bi-GRU(b^1, b^2, \dots, b^{|N(u)|}). \tag{7}$$

where  $h_b^i \in R^d$ .

We obtain the user’s multi-behavior preference sequence based on the user’s behavior information and interactive item information. As we all know, users’ way of thinking and external market conditions change over time. If the model does not pay attention to changes in the user’s core behavior, it will cause errors in subsequent recommendations. Inspired by Chang et al. [21], we input the user multi-behavior preference sequence into a GRU network with an attention update gate (AUGRU) to obtain the user’s final multi-behavior preference representation:

$$h_{u_b} = AUGRU(h_b^1, h_b^2, \dots, h_b^{|N(u)|}) \tag{8}$$

The AUGRU model uses an attention mechanism to process differentiated multi-behavior information. It scales the individual multi-behavior features of the update gates by using attention scores. Therefore, behavior features with less correlation have less influence on the hidden state. This makes the acquired multi-behavior information changes more accurate.

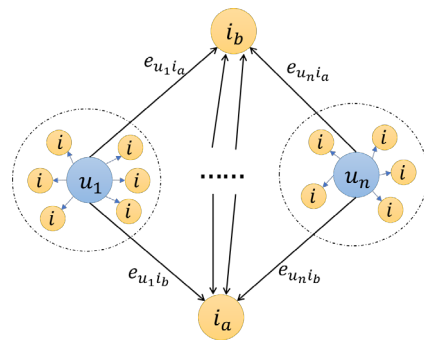
### 2.5. Item–Item Multi-Behavior Interaction Information Aggregation

#### 2.5.1. Construction of Item–Item Relationship Graph

Even for the same item, different users may show different meanings when interacting with the item. We can mine the connection between items from the perspective of users, and then obtain the potential representation of items. Therefore, we extract the item set with the same interaction type as the target item,  $i_j$ , in the user–item interaction graph,  $\mathcal{G}_0$  (Figure 2). Then we construct the item–item multi-behavior interaction graph,  $\mathcal{G}_i$ , for further learning the latent factors of items. The weight of each interactive edge is expressed as follows:

$$e_{jj'}^* = \sum_{m \in N_{\mathcal{G}_1}(j,j')} \alpha_{jm} \alpha_{j'm} e_{jj'} = \frac{\exp(e_{jj'}^*)}{\exp(\sum_{r \in N_{\mathcal{G}_1}(j)} e_{jr}^*)}, \tag{9}$$

where  $\alpha_{jm}$  and  $\alpha_{j'm}$  are the weight information calculated by Formula (1).  $N_{\mathcal{G}_1}(j, j')$  represents the users adjacent to  $j$  and  $j'$  in graph  $\mathcal{G}_1$ .  $N_{\mathcal{G}_1}(j)$  represents the items that are second-order adjacent to  $\mathcal{G}_1$  and  $j$  in the graph. The final attention weight  $e_{jj'}$  is obtained by normalizing  $e_{jj'}^*$  using the Softmax function.



**Figure 2.** The principal figure of item–item relation information aggregation.

### 2.5.2. Information Propagation of Item–Item Relationship Graph

Through the weight information, we can define the information propagation method of neighbor item  $j$  to item  $i$ . Based on the weight information of the item–relation graph and the feature information of neighbor items, we obtain the extended representation of item  $i$ :

$$h_{i_s} = f\left(\sum_{j \in N^i(i)} e_{ij} h_j^{(*)} + h_i\right), \tag{10}$$

where  $N^i(i)$  represents the neighborhood of  $i$  in the item–item interaction graph  $\mathcal{G}_i$ , and  $h_{i_s}$  is the aggregated information of  $i$ . Moreover,  $f(\cdot)$  is an activation function similar to LeakyReLU.

### 2.6. Joint Prediction Module

After the above three modules, we obtain the user’s preference behavior,  $h_{u_b}$ ; the user interest feature,  $h_u^{(*)}$ ; the clustering feature,  $h_{i_s}$ ; and the feature,  $h_i^{(*)}$ , of the item. We combine the above feature information to obtain the final embeddings of users and items for the final prediction:

$$h_u = h_u^{(*)} \oplus h_{u_b}, h_i = h_i^{(*)} \oplus h_{i_s}, \tag{11}$$

where  $\oplus$  represents addition between vector elements. Finally, we inner-product the final representations of users and items to predict their match scores:

$$\hat{y}_{(u,i)} = h_u^T \cdot h_i, \tag{12}$$

### 2.7. Model Learning

Given a user–item interaction sequence,  $S = \{s_1, s_2, \dots, s_{|S|}\}$ , we extract its top  $|S| - x$  items to predict the  $x$  items of its last interaction. To optimize our UMBGN model, we choose BPRloss [22], which is widely used in recommendation systems [9,17]. Specifically, the final loss function is denoted as follows:

$$\mathcal{L} = \sum_{(u,i^+,i^-) \in \mathcal{O}} -\ln \sigma(\hat{y}_{(u,i^+)} - \hat{y}_{(u,i^-)}) + \lambda \|\Theta\|_2^2, \tag{13}$$

where  $\mathcal{O} = \{(u, i^+, i^-) | (u, i^+) \in R^+, (u, i^-) \in R^-\}$  represents the paired target behavior training dataset;  $R^+$  and  $R^-$  refer to target behaviors that have occurred and target behaviors that have not occurred, respectively;  $\sigma$  refers to the sigmoid function;  $\Theta$  is a parameter that can be trained in the network; and  $\lambda$  is the L2 normalization coefficient.

## 3. Experiment

In this section, we recount the experiments we conducted on three real datasets, namely MovieLens, Yelp2018, and Online Mall, to evaluate our UMBGN model. We explore the following four questions:

RQ1: In this paper, we consider user multi-behavior information. Does this improve recommendation performance? How does UMBGN perform compared to existing models?

RQ2: We also set the propagation weight among network nodes according to the behavior information. Does this improve the performance of the model? If the weight information is not considered, what will be the effect on the experimental results?

RQ3: How does each module of the model contribute to the improvement of the accuracy of the prediction results?

RQ4: What are the effects of various parameters of the model on the final performance of our proposed method?

### 3.1. Experimental Environment

#### 3.1.1. Datasets

To evaluate the performance of UMBGN, we conduct experiments on MovieLens, Yelp2018, and the real e-commerce dataset Online Mall, respectively.

MovieLens is a widely used benchmark dataset in recommendation systems containing 20 million movie ratings (accessed at <https://grouplens.org/datasets/movielens/20m/>, accessed on 15 April 2022). In the experiment, we divided user ratings into multiple behavior types: (1) dislike behavior, (2) neutral behavior, and (3) like behavior.

Yelp2018 is a famous merchant-review website in the US (accessed at <https://www.yelp.com/dataset/download>, accessed on 16 April 2022). Users can rate merchants, submit reviews, and give tips on the Yelp website. We divided the Yelp dataset into four behaviors (like, dislike, neutral, and tip), using the same criteria as we did for MovieLens.

Online Mall is provided by JD.com, a commerce company with a huge number of users and a full range of goods (accessed at <https://jdata.jd.com/html/detail.html?id=8>, accessed on 16 April 2022). User-behavior types include click, favorite, add to cart, and purchase.

To ensure the accuracy of the experiments, we performed basic preprocessing on the dataset. We removed users and items with fewer than 10 interactions. Then we divide the dataset into the training set, validation set, and test set according to 80%, 10%, and 10%. The dataset information after data preprocessing is shown in Table 1.

**Table 1.** Experimental dataset statistics.

Dataset	User	Item	Interaction	Behavior Type
Yelp2018	31,668	38,048	1,561,406	{Tip, Dislike, Neutral, Like}
ML-20 M	138,493	26,744	19,989,593	{Dislike, Neutral, Like}
Online Mall	102,703	24,677	37,059,872	{Buy, Cart, Fav, PV}

#### 3.1.2. Comparison Methods

To evaluate our method, we adopted two evaluation metrics that were widely used in previous work: recall@K and NDCG@K [18]. They are defined as follows:

Recall@K: It is used to measure the probability that the actual interaction item appears in the top-K leaderboard recommendation task. Recall@K does not pay attention to the order in which the user actually clicks an item in the recommended task list; it only considers whether the item appears in the top N positions of the recommended task list.

NDCG@K: In the top-K ranking list, NDCG@K evaluates the quality of the recommendation list according to the rank order of correct items. It assigns higher scores to higher-ranked positions, which means that test items should be ranked as high as possible.

For each user in the test set, we adopt the next-item-recommendation task [23]. For each user, we pair the ground-truth items in the test set with other negative items that are interactions, obtain the user's preference score for all items, and then rank them. In this paper, we set K=10. It is known that higher HR and NDCG scores indicate a better model performance.



### 3.1.3. Parameter Setting

In this paper, we use TensorFlow to implement the UMBGN model and use the Adam optimizer to infer the model parameters. We performed experiments on two NVIDIA GeForce GTX2080 Ti GPUs. Firstly, we initialized the user–item embedding matrix and the weights of each item in the mixture model. The embedding dimension of users and items is set to 32. The initial learning rate and the batch size are set to 0.01 and 64, respectively. Secondly, a regularization strategy with weight decay selected from the set of {0.1, 0.05, 0.01, 0.005, 0.001} was used to alleviate the overfitting problem during the training phase. In our evaluation, we employed early stopping to terminate training when the performance on the validation data degraded for 5 consecutive epochs.

### 3.1.4. Baseline

To verify the effectiveness of the UMBGN model, we compare it with six baseline models: two traditional recommendation methods, two RNN-based methods, and two graph network recommendation methods. We briefly describe the six baseline models as follows:

BPR-MF [24]: It optimizes the latent factor of implicit feedback, using pairwise ranking loss in Bayesian methods to maximize the gap between positive and negative terms.

FPMC [25]: This is a classic mixed model that captures sequential effects and the general interest of users. FPMC fuses sequence and personalized information for recommendation by constructing a Markov transition matrix.

GRURec [19]: It is a GRU model trained based on a parallel mini-batch top1 loss function. GRURec uses parallel computation, as well as mini-batch computation, to learn model parameters.

GRU4Rec+ [26]: This is an improved version of GRURec, which concatenates the hot term vector and the feature vector as the input GRU network and has a new loss function and sampling strategy.

GraphRec [27]: It is a deep graph neural network model that enriches the information representation of nodes through embedding propagation and aggregation. GraphRec also aggregates social relations among users through a graph neural network structure.

NGCF [18]: It is an advanced graph neural network model. NGCF has some special designs that can combine traditional collaborative filtering with graph neural networks for application in recommendation systems.

Among all of these methods, BPR-MF and FPMC are traditional recommendation methods, GRURec and GRU4Rec+ are RNN-based methods, and GraphRec and NGCF are graph-network-based methods.

## 3.2. Performance Comparison

We demonstrate the performance of the above methods in predicting target types for user–item interactions on three real datasets. As shown in Table 2, UMBGN achieves significant performance improvement on different types of datasets. This improvement benefits from our consideration of the user’s multi-behavior interaction sequence and the relationship between items.

**Table 2.** Performance comparison of all methods in terms of Recall@10 and NDCG@10 on all datasets.

Dataset	Metric	BPR-MF	FPMC	GRURec	GRU4Rec+	NGCF	GraphRec	UMBGN
Yelp2018	Recall	0.2136	0.3811	0.6689	0.7051	0.7813	0.7594	0.7938
	NDCG	0.1208	0.2439	0.3607	0.4518	0.5232	0.4943	0.5362
ML-20 M	Recall	0.2608	0.3404	0.3748	0.4976	0.6020	0.5891	0.6351
	NDCG	0.1365	0.3163	0.2942	0.4129	0.5029	0.4982	0.5137
Online Mall	Recall	0.2679	0.3637	0.4474	0.6786	0.7801	0.7546	0.8293
	NDCG	0.1347	0.2692	0.3815	0.4641	0.5832	0.5317	0.5841

The experimental result shows that BPR-MF performed poorly overall. This may be because it cannot consider the user's long-term preference information. It proves that some traditional matrix factorization methods are not suitable for multi-behavior recommendation tasks. Although FPMC has an improved performance compared with BPR-MF, it still has not achieved satisfactory results. RNN-based models (GRURec, GRU4Rec+) have been greatly improved compared to traditional methods because RNN-based models can capture users' long-term preferences more effectively. In addition, GRU4Rec+ performs better than GRURec. This may be attributed to GRU4Rec+ considering personalized information.

Graph-network-based models (GraphRec, NGCF, and UMBGN) significantly outperform traditional methods and RNN-based methods. This shows that using the graph network method can better mine user-item connections and have a better ability to recommend the next item. Furthermore, we observe that UMBGN outperforms other datasets in the Online Mall dataset. One possible explanation is that Online Mall has a large amount of data and rich types of user-item interactions. In addition, the number of users in the Online Mall dataset is relatively large, thus enabling the model to better model user preference information. Therefore, UMBGN is more practical in the real world with massive user data, such as online shopping platforms and social platforms. This shows that considering the multi-behavior information of users improves the recommendation performance.

### 3.3. Ablation Experiments

#### 3.3.1. The Influence of Different Behavioral Weights on the Experimental Results

To evaluate the impact of different behavioral information on user purchase intention, we compared the performance of our method on the Online Mall dataset. We designed the following controlled experiments: (1) setting the behavior weight of each user to the same weight,  $\alpha_{uk}$ ; and (2) setting each interaction behavior to the same weight,  $w_k^u$ . We present the results of the ablation experiments in Table 3. It shows that our UMBGN model with learnable behavior weight information is 68.45% higher than the model with the same  $\alpha_{uk}$  and 34.10% higher than the model with the same  $w_k^u$  on recall@10. It is 50.27% higher than the model with the same  $\alpha_{uk}$  and 25.21% higher than the model with the same  $w_k^u$  on NDCG@10. This indicates that focusing on multi-behavior weights is necessary and should be learned by the model itself. Therefore, setting the propagation weights between graph network nodes according to multi-behavior information improves the performance of the model.

**Table 3.** Results of ablation experiments with multi-behavior weights on Online Mall.

Model	UMBGN	Same $\alpha_{uk}$	Same $w_k^u$
Recall@10	0.8293	0.4923	0.6182
NDCG@10	0.5841	0.3887	0.4665

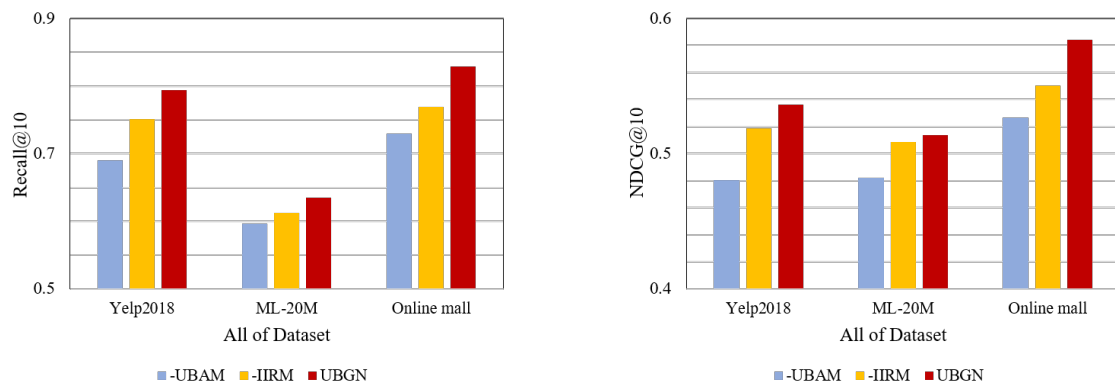
#### 3.3.2. The Influence of Each Module in UMBGN on the Experimental Results

The user multi-behavior awareness module aims to obtain the user's behavior preference information, and the item-information-relation module aims to obtain the relevant information between items. They are both complementary to the user-item interaction information module. We conducted an ablation study to test the effectiveness of the user multi-behavior awareness module and the item information relation module in our UMBGN. The results are shown in Table 4.

**Table 4.** Performance of user multi-behavior awareness module and item information relation module.

Dataset	Yelp2018		ML-20 M		Online Mall	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
UMBGN	0.7938	0.5362	0.6351	0.5137	0.8293	0.5841
w/o UBAM	0.6905	0.4803	0.5960	0.4821	0.7294	0.5267
w/o IIRM	0.7513	0.5189	0.6127	0.5088	0.7687	0.5502

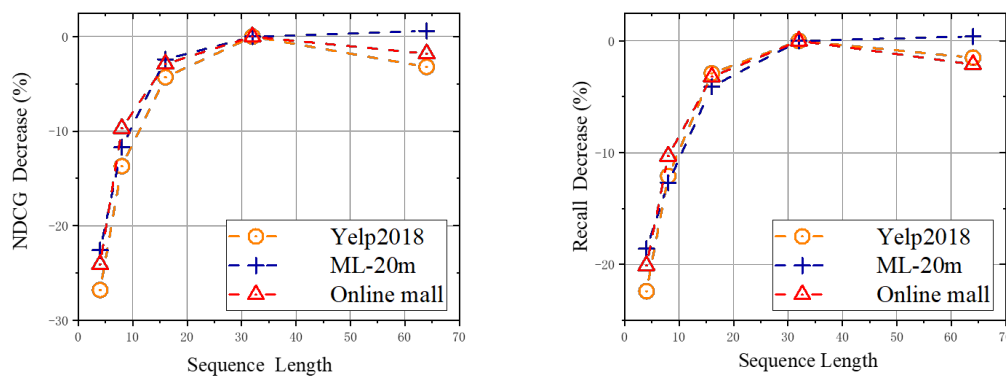
The results of the ablation experiments (Figure 3) show that the UMBGN model has a higher recall rate and NDCG than the model without the user multi-behavior awareness module and the item-information-relation model. Especially on the Online Mall dataset, it improves the recall rate by 13.70% and 7.88%, respectively. Moreover, it improves the NDCG by 9.83% and 5.80%, respectively. This shows that taking into account the user's multi-behavior interaction sequence and the relationship between items can make more accurate recommendations to users. This shows that each module is necessary to improve the accuracy of the prediction results.

**Figure 3.** Results of ablation experiments of sub-modules in UMBGN.

### 3.4. Parametric Analysis

#### 3.4.1. The Effect of Sequence Length on Prediction Results

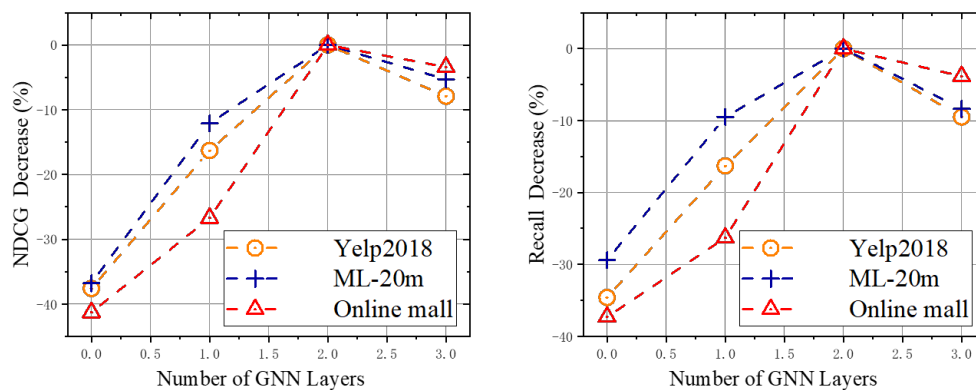
We also explored the effect of the maximum length,  $N$ , of user–item interaction sequences on the model recommendation performance. Figure 4 shows the impact of the maximum length,  $N$ , on the recommendation performance on the ML-20 m dataset and the Online Mall dataset, respectively. We observe that the recommendation performance improves as the  $N$  increases until the  $N$  is less than 40. This indicates that the length of the user's behavior sequence has an impact on the recommendation performance. However, when  $N$  exceeds 40, the recommendation performance on the ML-20 m dataset no longer increases significantly. Moreover, the recommendation performance of the Online Mall dataset has also declined. This suggests that the model does not always benefit from larger  $N$ , as larger  $N$  tends to introduce more noise. However, our model remains stable when the length  $N$  becomes larger. This also proves that our model can handle noisy behavioral sequence information well.



**Figure 4.** Performance comparison of methods with different behavior sequence lengths,  $N$ , on three datasets.

### 3.4.2. The Influence of the Number of Layers of the Graph Neural Network on the Prediction Results

We wish to test the effect of the number of layers of the GNN on the UMBGN model. In the user–item interaction information module, UMBGN, with two recursive message propagation layers, achieves the best results. This shows that it is essential to model higher-order relationships between items and features via GNNs. However, as shown in Figure 5, the performance starts to degrade as the depth of the graph model increases. This is because multiple embedded propagation layers may contain some noisy signals, resulting in over-smoothing [28]. This shows that determining the optimal parameters of the model through a large number of experiments is conducive to improving the performance of the model.



**Figure 5.** Performance comparison of methods with different numbers of GNN layers on three datasets.

## 4. Related Work

### 4.1. Recommendation Based on Graph Neural Network

In recent years, graph networks that can naturally aggregate node information and topology have attracted extensive attention. Especially in recommendation systems, the use of graph networks to mine user–item interaction data has achieved remarkable results [29–31]. Yang et al. [32] constructed a Hierarchical Attention Convolutional Network (HAGERec) combined with a knowledge graph. They exploited the high-order connectivity relationship of heterogeneous knowledge graphs to mine users' latent preferences. In addition, information aggregation was performed on user and item entities through local proximity and attention mechanisms. Gwadabe et al. [33] proposed a GNN-based recommendation model, GRASER, for the session-based recommendation. It used GNN to learn the sequential and non-sequential complex transformation relationship between items in each session, which improved the performance of the recommendation. Zhang et al. [34]

proposed a dynamic graph neural network (DGSR) for the sequential recommendation. It explicitly modeled the dynamic collaboration information between different user sequences in sequential recommendations. Therefore, it could transform the task of the next prediction in sequential recommendation into a link prediction between user nodes and item nodes in a dynamic graph. Fan et al. [27] designed a graph network framework (GraphRec) for the social recommendation. The method jointly captured users' purchase preferences from the user's social graph and the user-item interaction graph. The SURGE graph neural network frame proposed by Chang et al. [21] combined the sequential recommendation model and the graph neural network model. This method first integrated the different preferences in the user's long-term behavior sequence into the graph structure, and then it performed operations such as perception, propagation, and pooling of the graph network. It could dynamically extract the core interests of the current user from noisy user behavior sequences. Different from their work, our work defines new multi-behavior information weights for information propagation in graph neural networks.

#### 4.2. Multi-Behavior Recommendation

Traditional recommendation systems usually rely only on a single type of user-item interaction, which limits the performance of the methods. Recommendation methods utilizing multiple behaviors can more accurately capture user preference information. Guo et al. [20] designed a Deep Intent Prediction Network (DIPN) to predict users' purchase intentions from multiple perspectives. They combined touch interaction behavior with traditional browsing behavior and introduced multi-task learning to differentiate user behavior. Experiments on large-scale datasets showed that the network significantly outperforms traditional methods that used only browsing interaction behavior. Rosaci [35,36] proposed a CILIOS method to determine inter-ontology similarities between agents. It monitored user behavior and interests to extend the recommendation dataset generated by traditional methods. In addition, this method extracted logical knowledge in recommendation scenarios to support web recommendations. Wu et al. [37] constructed a new multi-behavior multi-view contrastive learning recommendation model (MMCLR) to solve the data sparsity and cold-start problems in traditional recommender models. They considered the similarities and differences between different user behaviors and views through three tasks. Experiments on real datasets indicate that MMCLR significantly improved the performance of recommendations. Pan et al. [38] designed a Spatiotemporal Interaction Augmented Graph Neural Network (SIGMA). It encoded a mobile graph to represent individual mobile behavior and used a stacked scoring approach to generate recommendation scores. This showed that the mobile behavior of individuals and groups played an important role in location recommender systems. Xia et al. [39] developed a Multi-Behavior Graph Meta Network (MB-GMN) to extract the interaction information of multiple behavior types between users and items. The proposed method jointly models behavioral heterogeneity and interaction behavioral diversity, combined with the meta-learning paradigm. A large number of comparative experiments on three datasets demonstrated the effectiveness of their method. Inspired by the above research work, we propose a new multi-behavior awareness module to further mine time-series based user multi-behavior information.

## 5. Conclusions

In this paper, we explored the problem of graph network recommendation, focusing on user multi-behavior interaction sequences, and proposed a UMBGN model. Compared with the traditional GNN model, our model updates the node connection weights of the user-item interaction graph according to the multi-behavior interaction information, so that it can capture the user's interest in specific items under different behavioral information. In this study, we designed two modules to further mine the user's multi-behavior preference information. Firstly, we put the multi-behavior sequence information of the target user into an improved Bi-GRU model, the AUGRU model, to enrich the user's embedding representation. Secondly, we built an item-item graph based on the user's dependencies on items to

further enrich the embedding representation of items. The comparative experiments that we performed on three real datasets demonstrate the effectiveness of the UMBGN model. Further ablation experiments prove the necessity of the user multi-behavior awareness module and item information awareness module in our UMBGN model. In addition, we also evaluated the impact of different parameters on recommendation performance, confirming the applicability of UMBGN in practical applications. However, our approach does not consider potential connections among users. In the future, we plan to introduce users' social relations into our method to improve the accuracy of the next-item recommendation.

**Author Contributions:** Conceptualization, M.J. and F.L.; methodology, M.J. and X.Z.; software, M.J. and X.L.; validation, X.Z.; investigation, M.J.; resources, F.L.; data curation, M.J.; writing—original draft preparation, M.J. and X.L.; writing—review and editing, F.L. and X.Z.; visualization, M.J.; supervision, F.L.; funding acquisition, F.L.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded by the National Natural Science Foundation of Shandong (ZR202011 020044) and the National Natural Science Foundation of China (61772321).

**Data Availability Statement:** Publicly available datasets were analyzed in this study. The data of MovieLens can be found here: <https://grouplens.org/datasets/movielens/20m/> (accessed on 15 April 2022). The data of Yelp2018 can be found here: <https://www.yelp.com/dataset/download> (accessed on 16 April 2022). The data of Online Mall can be found here: <https://jdata.jd.com/html/detail.html?id=8> (accessed on 16 April 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Gu, Y.; Ding, Z.; Wang, S.; Yin, D. Hierarchical user profiling for e-commerce recommender systems. In Proceedings of the 13th International Conference on Web Search and Data Mining, Houston, TX, USA, 3–7 February 2020; pp. 223–231.
2. Schafer, J.B.; Frankowski, D.; Herlocker, J.; Sen, S. Collaborative filtering recommender systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 291–324.
3. Koren, Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 426–434.
4. Ning, X.; Karypis, G. Slim: Sparse linear methods for top-n recommender systems. In Proceedings of the 2011 IEEE 11th international conference on data mining, Vancouver, BC, Canada, 11–14 December 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 497–506.
5. Rendle, S.; Gantner, Z.; Freudenthaler, C.; Schmidt-Thieme, L. Fast context-aware recommendations with factorization machines. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, Beijing, China, 25–29 July 2011; pp. 635–644.
6. He, X.; Chua, T.S. Neural factorization machines for sparse predictive analytics. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 355–364.
7. 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, Perth, Australia, 3–7 April 2017; pp. 173–182.
8. Xue, H.J.; Dai, X.; Zhang, J.; Huang, S.; Chen, J. Deep matrix factorization models for recommender systems. In Proceedings of the IJCAI International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 3203–3209.
9. Fan, S.; Zhu, J.; Han, X.; Shi, C.; Hu, L.; Ma, B.; Li, Y. Metapath-guided heterogeneous graph neural network for intent recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2478–2486.
10. Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; Tan, T. Session-based recommendation with graph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 346–353.
11. 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, Anchorage, AK, USA, 4–8 August 2019; pp. 950–958.
12. Yang, L.; Liu, Z.; Dou, Y.; Ma, J.; Yu, P.S. Consisrec: Enhancing gnn for social recommendation via consistent neighbor aggregation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Montreal, QC, Canada, 11–15 July 2021; pp. 2141–2145.

13. Gao, C.; Zheng, Y.; Li, N.; Li, Y.; Qin, Y.; Piao, J.; Quan, Y.; Chang, J.; Jin, D.; He, X.; et al. Graph neural networks for recommender systems: Challenges, methods, and directions. *arXiv* **2021**, arXiv:2109.12843, 2021. [[CrossRef](#)]
14. Fan, Z.; Liu, Z.; Zhang, J.; Xiong, Y.; Zheng, L.; Yu, P.S. Continuous-time sequential recommendation with temporal graph collaborative transformer. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Gold Coast, Australia, 1–5 November 2021; pp. 433–442.
15. Zhao, Y.; Ou, M.; Zhang, R.; Li, M. Attributed Graph Neural Networks for Recommendation Systems on Large-Scale and Sparse Graph. *arXiv* **2021**, arXiv:2112.13389.
16. Tan, Q.; Zhang, J.; Yao, J.; Liu, N.; Zhou, J.; Yang, H.; Hu, X. Sparse-interest network for sequential recommendation. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining, Jerusalem, Israel, 8–12 March 2021; pp. 598–606.
17. Zheng, Y.; Gao, C.; He, X.; Li, Y.; Jin, D. Price-aware recommendation with graph convolutional networks. In Proceedings of the 2020 IEEE 36th International Conference on Data Engineering (ICDE), Dallas, TX, USA, 20–24 April 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 133–144.
18. Wang, X.; He, X.; Wang, M.; Feng, F.; Chua, T.-S. Neural graph collaborative filtering. In Proceedings of the SIGIR, Paris, France, 21–25 July 2019; pp. 165–174.
19. Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; Tikk, D. Session-based recommendations with recurrent neural networks. *arXiv* **2015**, arXiv:1511.06939.
20. 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, Anchorage, AK, USA, 4–8 August 2019; pp. 1984–1992.
21. Chang, J.; Gao, C.; Zheng, Y.; Hui, Y.; Niu, Y.; Song, Y.; Jin, D.; Li, Y. Sequential recommendation with graph neural networks. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Montreal, QC, Canada, 11–15 July 2021; pp. 378–387.
22. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* **2012**, arXiv:1205.2618.
23. Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; Jiang, P. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1441–1450.
24. Guo, H.; Tang, R.; Ye, Y.; Li, Z.; He, X. DeepFM: A factorization-machine based neural network for CTR prediction. *arXiv* **2017**, arXiv:1703.04247.
25. Rendle, S.; Freudenthaler, C.; Schmidt-Tieme, L. Factorizing personalized Markov chains for next-basket recommendation. In Proceedings of the 19th International Conference on World Wide Web-WWW'10, Raleigh, NC, USA, 26–30 April 2010.
26. Hidasi, B.; Karatzoglou, A. Recurrent neural networks with top-k gains for session-based recommendations. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 843–852.
27. Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, E.; Tang, J.; Yin, D. Graph neural networks for social recommendation. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; ACM: New York, NY, USA, 2019; pp. 417–426.
28. Chen, D.; Lin, Y.; Li, W.; Li, P.; Zhou, J.; Sun, X. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 3438–3445.
29. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
30. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [[CrossRef](#)] [[PubMed](#)]
31. Yin, R.; Li, K.; Zhang, G.; Lu, J. A deeper graph neural network for recommender systems. *Knowl. Based Syst.* **2019**, *185*, 105020. [[CrossRef](#)]
32. Yang, Z.; Dong, S. HAGERec: Hierarchical attention graph convolutional network incorporating knowledge graph for explainable recommendation. *Knowl. Based Syst.* **2020**, *204*, 106194. [[CrossRef](#)]
33. Gwadabe, T.R.; Liu, Y. Improving graph neural network for session-based recommendation system via non-sequential interactions. *Neurocomputing* **2022**, *468*, 111–122. [[CrossRef](#)]
34. Zhang, M.; Wu, S.; Yu, X.; Liu, Q.; Wang, L. Dynamic graph neural networks for sequential recommendation. *IEEE Trans. Knowl. Data Eng.* **2022**. [[CrossRef](#)]
35. Rosaci, D. Web Recommender Agents with Inductive Learning Capabilities. In *Emergent Web Intelligence: Advanced Information Retrieval*; Springer: London, UK, 2010; pp. 233–267. [[CrossRef](#)]
36. Rosaci, D. CILIOS: Connectionist inductive learning and inter-ontology similarities for recommending information agents. *Inf. Syst.* **2007**, *32*, 793–825. [[CrossRef](#)]
37. Wu, Y.; Xie, R.; Zhu, Y.; Ao, X.; Chen, X.; Zhang, X.; Zhuang, F.; Lin, L.; He, Q. Multi-view Multi-behavior Contrastive Learning in Recommendation. In Proceedings of the International Conference on Database Systems for Advanced Applications, Virtual Event, 11–14 April 2022; Springer: Cham, Switzerland, 2022; pp. 166–182.

38. Pan, X.; Cai, X.; Song, K.; Baker, T.; Gadekallu, T.R.; Yuan, X. Location recommendation based on mobility graph with individual and group influences. *IEEE Trans. Intell. Transp. Syst.* **2022**. [[CrossRef](#)]
39. Xia, L.; Xu, Y.; Huang, C.; Dai, P.; Bo, L. Graph meta network for multi-behavior recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Montreal, QC, Canada, 11–15 July 2021; pp. 757–766.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.