# MB-DP: A Multi-behavior Recommendation Model Integrating Dynamic Preferences

Cairong Yan, Xiaopeng Guan, Haixia Han, Zhaohui Zhang
School of Computer Science and Technology, Donghua University
{cryan, zhzhang}@dhu.edu.cn, {xpguan, haixiahan}@mail.dhu.edu.cn

***Abstract*—Multi-behavior recommendation has gained significant attention in recent years for its ability to outperform single-behavior models. Current research related to multi-behavior models leaves room for improvement in the following two areas. First, the noise carried by individual behaviors and the additional noise generated during behavior processing is often overlooked, and these can ultimately degrade recommendation performance. Second, the specific time period of behavioral interactions and the frequency of interactions within that time period are also not taken into account. To address the above limitations, we propose a multi-behavior recommendation model integrating dynamic preferences (MB-DP) that captures dynamic interests while smoothing and denoising multi-behavior information. MB-DP extracts low and high-order semantics from various behaviors and unifies the measurements to generate interaction predictions. Additionally, it analyzes the interaction time and frequency of each behavior using gated recurrent units to capture the dynamic preferences of users and improve the prediction values. Extensive experimental results on two real-world datasets show that MB-DP significantly improves recommendation performance compared to the state-of-the-art baselines.**

***Keywords-Recommender System; Multi-behavior; Dynamic Preference; Graph Neural Network; Gated Recurrent Unit***

## I. Introduction

With the explosive growth of information, users struggle to make decisions among countless options [1]. Recommender systems based on user behavior modeling are a good solution to this problem in various fields.

The use of behavioral data has increased in fields such as e-commerce and social networks [2]. However, single interaction types inadequately represent user interests, as interactions are often multi-typed. Multi-behavior recommendation has become a key research direction in recommender systems. Current models include deep learning-based and graph neural network-based approaches. Deep learning-based models consider the cascading relationship between behavior types and use multi-task learning for optimization [3]. Graph neural network-based models, like MB-GCN and GHCF, mine high-order interaction information between users and items, providing diverse content and structural information [4, 5, 6].

Multi-behavior recommendation models face two key issues: 1) existing models don't address noise impact in multi-behavior information on recommendation performance, leading to over-smoothing and affecting personalization [7]; 2) existing models do not consider users' dynamic preference information. As shown in Figure 1, in a multi-behavior recommendation model that does not consider dynamic preferences, two users have the same interaction behavior type for $i_1$, so similar preference degrees will be judged. However, by considering dynamic preferences over three time periods, we can see that only user $u_1$ interacts with item $i_1$ in the third period, and $u_1$ has more interactions with $i_1$ in the first period, indicating that $u_1$'s preference for $i_1$ is stronger than $u_2$'s. In real-world scenarios, users' preferences [8] and item popularity [9] can change over time, and capturing changing preferences requires factoring in time to capture dynamic preferences from users' multi-behavior.

Therefore, to address the aforementioned problems, in this paper, we propose a novel multi-behavior recommendation model integrating dynamic preferences (MB-DP). The goal of MB-DP is to establish a recommender system model based on multi-behavior learning and optimized prediction values using dynamic preferences. Specifically, We learn meta-knowledge from behavior interaction patterns, obtaining low and high-order personalized semantics. To reduce noise, we normalize initial measurements and use multi-layer graph neural networks. MB-DP analyzes time periods and interaction frequency to capture users' dynamic preferences and optimize recommendation results. The contributions of our paper are as below:

(1) We propose MB-DP, a new recommendation network that considers multi-behavior patterns and dynamic preferences, and extracts personalized semantics while smoothing out noise from multiple behaviors.

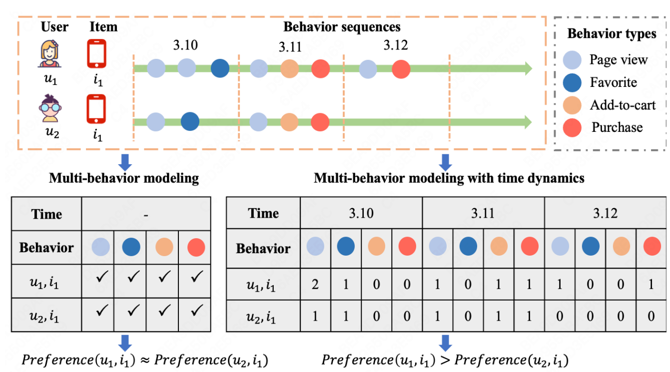


Figure 1. User preferences relate to multi-behavior types. A time-dynamic multi-behavior modeling approach captures more information.

This work is supported by Shanghai Science and Technology Innovation Action Plan Project (No. 22511100700)
DOI: 10.18293/SEKE23-130

(2) We use gated recurrent units to analyze interaction time and frequency, and correct multi-behavior learning results towards optimized prediction values.

(3) We show the effectiveness of MB-DP on two real-world datasets and conduct ablation studies to understand the model design. MB-DP outperforms baselines from various studies in terms of recommendation performance.

The paper is structured as follows: Section 2 reviews related work; Section 3 defines the research problem; Section 4 details our MB-DP model; Section 5 presents experimental results; Section 6 concludes our work and discusses future research.

## II. Related Work

The use of graph learning methods can uncover higher-order connectivity between users and items, with the core idea of enhancing node representation using (higher-order) neighborhoods [10]. NGCF [11] employs multi-layer graph neural networks for higher-order information aggregation, while LightGCN [12] preserves user and item embeddings to prevent over-smoothing. Zhang et al. proposed a new second-order continuous GNN with better interpretability and avoided over-smoothing [7]. GAMLP [13] captures potential correlations between different scale graph knowledge, while InstantGNN [14] proposes an incremental calculation method for graph representation matrices to address continuous changes in large-scale dynamic graphs.

Multi-behavior recommendation models treat added behaviors as auxiliary compared to single-behavior models. NMTR [3] shares user and item embedding layers across behavior types. MB-GCN [5] unifies user-item interaction matrices into a graph, capturing behavior strength and semantics through propagation layers. MB-GMN [6] uses meta-learning to incorporate multiple behavior patterns and learn behavior representations related to behavior types. GHCF [4] jointly embeds node representations and relationships into multi-relational prediction using GCN and performs non-sampling optimization under multi-task learning. S-MBRec [15] performs GCN for each behavior, differentiating the importance of different behaviors through a supervised task, and capturing the commonality of embeddings between target and auxiliary behaviors with a star-shaped contrastive learning task. However, existing multi-behavior recommendation models often overlook the positive impact of users' dynamic preferences on recommendations. MB-DP addresses this issue by designing a dynamic preference module.

Although the above multi-behavior recommendation models consider the impact of multiple behaviors on personalized recommendations, they do not consider the important time factor in real-world scenarios, where customer preferences for items change over time [16]. SASRec [17] identifies relevant items from user history to predict the next item in each time period. FISSA [18] improves SASRec by incorporating a global representation learning module that balances local and global representations using candidate item information. DUMN [19] designs a user memory network to model long-term interests in a granular way. Considering the influence of time length, URPI-GRU [20] learns users' short-term preferences through a GRU model and uses K-nearest neighbor sequence mining to explore

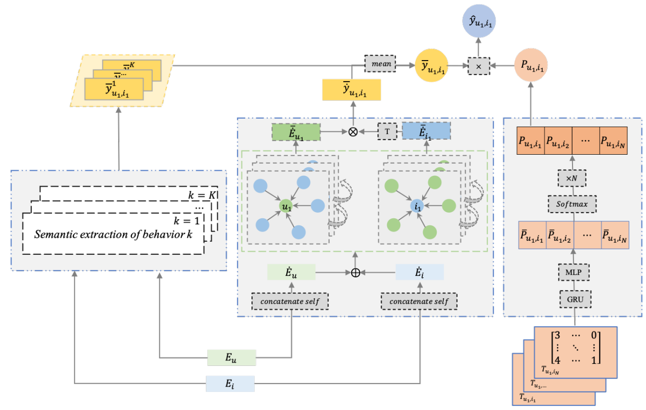


Figure 2. The architecture of MB-DP. It consists of three core modules, from left to right, behavior semantic extraction, behavior relationship denoising, and dynamic preferences extraction

users' long-term preferences. However, these models typically do not sufficiently extract high-order content and structural information from interactions.

## III. Problem Definition

We represent the set of users and items with $U = \{u_1, u_2, ..., u_M\}$ and $I = \{i_1, i_2, ..., i_N\}$, respectively. To capture multi-behavior interactions, we use a three-dimensional matrix $X \in R^{K\times M\times N}$ where $x_{u,i}^k = 1$ if there is an interaction of the $k$-th behavior category between user $u$ and item $i$, and 0 otherwise. To capture temporal information, we use a four-dimensional matrix $T \in R^{M\times N\times D\times K}$ where $t_{u,i}^{d,k} = z$ represents the number of interactions of the $k$-th behavior category between user $u$ and item $i$ during the $d$-th time period. We formulate the problem as predicting the probability of user $u_m$ performing the target behavior on item $i_n$ using $X$ and $T$ as inputs.

## IV. Methodology

We now introduce the details of our proposed MB-DP model, as shown in Figure 2. It consists of three core modules.

### A. Behavior Semantic Extraction Module

Different behaviors represent various user preferences, making some irrelevant interactions act as noise. Previous multi-behavior models focus on preference differences, while our goal is to extract user-specific behavior semantics.

#### 1) Low-order Personalized Semantic Extraction

We use a method inspired by MB-GMN [6] to learn meta-knowledge in multi-behavior interaction patterns and obtain personalized user and item embeddings with behavior semantic information. To obtain the initial contextualized user embedding $\widetilde{H}_u^k \in R^{M\times 3dim}$, we concatenate the behavior embedding $E_b^k \in R^{M\times dim}$ of the user under behavior $k$, the user embedding $E_u \in R^{M\times dim}$, and the project interaction information embedding $E_{u,N}^k$ in the second dimension:

$$\widetilde{H}_u^k = E_b^k \oplus E_u \oplus E_{u,N}^k, \tag{1}$$

where $E_{u,N}^k \in R^{M\times dim}$ is obtained by normalizing the interaction matrix $R^k \in R^{M\times N}$ and performing a cross item with the project embedding $E_i \in R^{N\times dim}$.

To balance the degree of information acquisition of the contextualized user embedding and the user embedding $E_u$ under behavior $k$, we pass the initial contextualized user embedding $\widetilde{H}_u^k$ through a linear layer with an input length of $3dim$, an output length of $dim$, and apply $LeakyReLU$ for non-linear activation. This yields the contextually balanced user embedding $H_u^k \in R^{M\times dim}$:

$$H_u^k = LeakyReLu\left(MLP\left(\widetilde{H}_u^k\right)\right). \tag{2}$$

To address the problem of low efficiency in meta-learning caused by high dimensionality, we use low-rank decomposition to transform meta-learning by learning two low-rank projections, $\overline{V}_u^{k,1} \in R^{M\times dim\times rank}$ and $\overline{V}_u^{k,2} \in R^{M\times rank\times dim}$ for behavior $k$, and obtain the contextualized user embedding $\widetilde{\mathrm{E}}_{\mathrm{ub}}^{\mathrm{k}}$ as follows:

$$\tilde{E}_{ub}^k = operate\left(operate\left(E_u, \bar{V}_u^{k,1}\right)^k, \bar{V}_u^{k,2}\right),$$
$$operate(a,b) = sum(unsqueeze(a) \odot b). \tag{3}$$

Here, $\overline{V}_u^{k,1}$ and $\overline{V}_u^{k,2}$ are obtained by applying a Linear layer with input length $dim$ and output length $rank \times dim$ to the contextually embedded user embedding $H_u^k$ after information balancing, and then reshaping the resulting tensor into $R^{M\times dim\times rank}$ and $R^{M\times rank\times dim}$ respectively.

By concatenating $\widetilde{E}_{ub}^k$ with $E_u$, we obtain the user embedding $E_{ub}^k \in R^{M\times 2dim}$ with personalized behavioral semantic information.

*2) High-order Connected Semantic Extraction Module*

We use graph neural networks to capture the similarity between users or items influenced by different behaviors.

Under behavior $k$, we define the adjacency matrix $A^k \in R^{(M+N)\times(M+N)}$ as follows:

$$A^k = \begin{pmatrix} 0 & X^k \\ X^{k^T} & 0 \end{pmatrix}. \tag{4}$$

We use the symmetric normalization matrix $\widetilde{A}^k \in R^{(M+N)\times(M+N)}$ to smooth the input matrix that needs to be processed, and the symmetric normalization matrix is defined as follows:

$$\widetilde{A}^k = D_k^{-\frac{1}{2}} A^k D_k^{-\frac{1}{2}}. \tag{5}$$

Here, $D_k$ is a diagonal matrix of size $(M+N) \times (M+N)$, and the values on the diagonal of $D_k$ represent the number of non−zero elements in each row of the adjacency matrix $A^k$. The input matrix $E^{k,(0)} \in R^{(M+N)\times 2dim}$ of the graph neural network at layer 0 is obtained by concatenating $E_{ub}^k$ and $E_{ib}^k$, which have personalized semantic information for behavior $k$.

The propagation rule for each layer of the graph neural network is:

$$E^{k,(l+1)} = \widetilde{A}^k \otimes E^{k,(l)}. \tag{6}$$

Finally, the final graph neural network output embedding matrix is obtained as follows:

$$E^k = mean\left(E^{k,(0)}, E^{k,(1)}, \ldots, E^{k,(L)}\right). \tag{7}$$

We split $E^k$ into two embedding matrices according to $[M, N]$, representing the final user embedding $\overline{E}_u^k \in R^{M\times 2dim}$ and item embedding $\overline{E}_i^k \in R^{N\times 2dim}$ respectively under behavior $k$:

$$\overline{E}_u^k, \overline{E}_i^k \longleftarrow split\left(E^k, [M, N]\right). \tag{8}$$

*B. Behavior Relationship Denoising Module*

The initial embeddings of each behavior cannot be uniformly scaled, which leads to different measurement scales of the high−order personalized semantics carried by each behavior type. In order to reduce the noise brought by multi-behavior interaction, we additionally treat all types of behaviors as a new behavior. To maintain the consistent shape with the high-order semantic output of each behavior, we concatenate the initial user embedding $E_u \in R^{M\times dim}$ and item embeddings $E_i \in R^{M\times dim}$ to themselves, resulting in new user embedding $\dot{E}_u \in R^{M\times 2dim}$ and new item embedding $\dot{E}_i \in R^{M\times 2dim}$:

$$\dot{E}_u = E_u \oplus E_u, \dot{E}_i = E_i \oplus E_i. \tag{9}$$

After performing high-order connected semantic extraction on them, we obtain $\overline{\dot{E}}_u \in R^{N\times 2dim}$ and $\overline{\dot{E}}_i \in R^{N\times 2dim}$.

*C. Dynamic Preferences Extraction Module*

Most multi-behavior models that use graph neural networks consider only the interaction adjacency matrix between users and items, which ignores the varying user preferences for an item based on behavior frequency and order. To address this, we propose a dynamic preferences extraction module to capture the occurrence frequency of each behavior interaction and analyze dynamic preferences for optimizing prediction results.

Using a one-layer unidirectional GRU with ReLU activation function, we extract preferences information for each behavior interaction in each time period, analyze the changes in preference levels, and obtain $\overline{T}_{\bar{u},\bar{i}} \in R^{d_{gruout}}$, where $d_{gruout}$ is the length of current time period information output by this unidirectional GRU.

$$\overline{T}_{\bar{u},\bar{i}} = \mathrm{ReLu}\left(\mathrm{GRU}\left(T_{\bar{u},\bar{i}}\right)\right). \tag{10}$$

To obtain the initial preference coefficient $\overline{P}_{\bar{u},\bar{i}}$ for user $\bar{u}$ and item $\bar{i}$, we pass $\overline{T}_{\bar{u},\bar{i}}$ through a linear layer with input length $d_{gruout}$ and output length 1.

To improve the robustness of the model, we optimize the initial preferences coefficients based on comparisons of $\overline{P}_{\bar{u}}$ for all items. The resulting final preferences coefficient matrix $P \in R^{M\times N}$ represents user $\bar{u}$'s preferences for item $\bar{i}$, where $\overline{P}_{\bar{u},\bar{i}}$ is the coefficient. We first aggregate initial coefficients for all items using a Softmax layer to obtain $\overline{P}_{\bar{u}} \in R^N$ for user $\bar{u}$. To preserve diversity, we multiply $\overline{P}_{\bar{u}}$ by $N$ to obtain the final preferences coefficient of user $\bar{u}$ for all items:

$$P_{\bar{u}} = Softmax(\overline{P}_{\bar{u}}) * N. \tag{11}$$

### D. Model Training

Inspired by LightGCN [12], for model training using Mini-Batch method, the trainable parameters are divided into two parts: 1) the initial user and item embeddings $\Phi = E^{(0)}$ that are affected by the Mini−Batch size, and 2) the other parameters $\Theta$. We use the Bayesian Personalized Ranking (BPR) loss, which is a pairwise loss that maximizes the difference between the positive samples (items with high user ratings) and negative samples (items with low user ratings) to improve the effectiveness of the recommender system. During the training phase, we optimize the following objective defined by using the Adam algorithm:

$$L_{BPR} = -\sum_{\bar{u}=1}^{M}\sum_{\bar{i}\in N_u}\sum_{\bar{j}\in N_u} ln\sigma(\hat{y}_{\bar{u}\bar{i}} - \hat{y}_{\bar{u}\bar{j}}) + \lambda\left(\left\|E^{(0)}\right\|_F^2 + \|\Theta\|_F^2\right), \tag{12}$$

where λ is the coefficient controlling L2 regularization and $\hat{y}_{\bar{u}\bar{i}}$ represents the final preferences prediction value of user $\bar{u}$ for item $\bar{i}$. Its calculation formula is:

$$\hat{y}_{\bar{u}\bar{i}} = \bar{y}_{\bar{u},\bar{i}} \times I_{\bar{u},\bar{i}} = \frac{\overline{E}_u^k \otimes \overline{E}_i^{k^T} + \sum_{k=0}^{K}\overline{E}_u^k \otimes \overline{E}_i^{k^T}}{K+1} \times P_{\bar{u},\bar{i}}. \tag{13}$$

## V. Experiments

This section conducts experiments on two real-world datasets to evaluate the performance of our MB-DP by comparing it with various recommendation techniques. Our goal is to answer the following questions:

- **RQ1**: How does MB-DP perform when competing with various recommendation baselines?
- **RQ2**: How do the sub-modules in MB-DP affect its recommendation performance?
- **RQ3**: How does different configurations of key hyperparameters affect the performance of MB-DP?

### A. Experimental Settings

***Datasets***. We evaluate our model on two real-world datasets: UserBehavior Data from Taobao, an e-commerce platform in China, and IJCAI Data from Tmall, a shopping platform. UserBehavior Data includes four types of user-item interactions and corresponding timestamps, while IJCAI Data contains the shopping logs of anonymous users before and after "Double Eleven". We use the purchase behavior as the target behavior and summarized the dataset information in Table I.

***Evaluation Protocols***. For performance evaluation in top-N recommendation tasks, we use two widely adopted metrics: Normalized Discounted Cumulative Gain (NDCG@10) and Hit Rate (HR@10). We use leave-one-out evaluation for training and test set split, and reserve the first interaction after the last time period as the test dataset. We randomly select 99 uninteracted items for each user as negative examples.

TABLE I. THE PREPROCESSED DATASETS

| Dataset | User# | Item# | Interaction# | Behavior types |
|---|---|---|---|---|
| UserBehavior | 38042 | 28747 | 599977 | {pv, fav, cart, buy} |
| IJCAI | 11854 | 21356 | 184995 | {pv, fav, cart, buy} |

***Methods for Comparison***. To comprehensively evaluate the performance, we evaluate the performance of MB-DP by comparing it with various baselines from different research directions, including MF, NGCF [11], LightGCN [12], NGCF-M, LightGCN-M, and MB-GMN [6]. MF is a traditional matrix factorization approach. NGCF is a model that uses graph neural networks to generate node embeddings by propagating embeddings over the user-item bipartite graph, allowing for higher-order information aggregation. LightGCN is an improvement on GCN and is used in recommender systems to address training and prediction on large-scale graphs. NGCF-M and LightGCN-M treat all interaction behaviors as one and train using the NGCF and LightGCN models, respectively. MB-GMN is a model that uses graph meta-network technology to solve the problem of multi-behavior recommendation, learning the relationships between users and items using graph meta-networks and using attention mechanisms to handle relationships between different behaviors.

***Parameter Settings***. We use PyTorch to implement the MB-DP model, and optimize the parameters using the Adam optimizer with a learning rate of $1e^{-2}$. The regularization coefficient $\lambda$ is chosen from {0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001}, and the batch size used during training is chosen from {32, 128, 256, 512, 1024, 2048}. By default, the hidden dimension of MB-DP is set to 64, the low-rank dimension is set to 4, and the hidden dimension of $GRU$ is set to 16. The number of graph neural network layers in MB-DP is searched from {1, 2, 3, 4}. The neural network baselines used in the experiments are either implemented using their original code or with the above parameter settings.

### B. Performance Validation (RQ1)

We evaluate the performance of predicting target user interaction behavior on different datasets and summarized the following observations:

Table II displays the performance of recommending top-$N$ items for different target behavior types. From the results presented, we observe that MB-DP consistently achieves the best performance in top-N recommendations for different N values. This indicates that MB-DP has an advantage over other baselines in providing the correct interactive items and has a higher probability of doing so. Such performance differences can be attributed to its effective extraction of user multi-behavior information and dynamic preferences.

Our evaluation shows that incorporating user-item multi-behavior information into recommendation models improves performance, particularly with GNN-based models that capture higher-order relationships through stacked information propagation layers. We also find that MB-DP outperforms several multi-behavior recommendation model baselines, including MB-GMN and LightGCN, by effectively extracting dynamic preferences for multi-behavior. The significant

TABLE II. OVERALL MODEL PERFORMANCE ON USERBEHAVIOR AND IJCAI DATASETS, WITH THE METRICS OF HR@$N$ AND NDCG@$N$

| Dataset | Method | HR@N | | | | NDCG@N | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | *N=1* | *N=5* | *N=10* | *N=20* | *N=1* | *N=5* | *N=10* | *N=20* |
| UserBehavior | MF | 0.0766 | 0.1528 | 0.2000 | 0.2891 | 0.0766 | 0.1175 | 0.1325 | 0.1549 |
| | NGCF | 0.1280 | 0.1975 | 0.2566 | 0.3539 | 0.1280 | 0.1630 | 0.1819 | 0.2063 |
| | lightGCN | 0.1313 | 0.1907 | 0.2453 | 0.3386 | 0.1313 | 0.1613 | 0.1788 | 0.2021 |
| | NGCF-M | 0.1365 | 0.3419 | 0.4691 | 0.6153 | 0.1365 | 0.2417 | 0.2827 | 0.3197 |
| | lightGCN-M | **0.3505** | **0.4861** | **0.5499** | **0.6375** | **0.3505** | **0.4229** | **0.4434** | **0.4655** |
| | MB-GMN | 0.1357 | 0.2555 | 0.3506 | 0.4852 | 0.1357 | 0.1965 | 0.2280 | 0.2610 |
| | **MB-DP** | **0.3944** | **0.5014** | **0.5635** | **0.6485** | **0.3944** | **0.4506** | **0.4705** | **0.4919** |
| | **Improvement** | **12.53%** | **3.15%** | **2.47%** | **1.74%** | **12.53%** | **6.53%** | **6.11%** | **5.68%** |
| IJCAI | MF | 0.0105 | 0.0649 | 0.1151 | 0.2133 | 0.0105 | 0.0372 | 0.0534 | 0.0780 |
| | NGCF | 0.0366 | 0.1148 | 0.2014 | 0.3295 | 0.0366 | 0.0749 | 0.1026 | 0.1347 |
| | lightGCN | 0.0424 | 0.1095 | 0.1775 | 0.2751 | 0.0424 | 0.0749 | 0.0967 | 0.1213 |
| | NGCF-M | 0.1936 | 0.4326 | 0.5683 | 0.7073 | 0.1936 | 0.3157 | 0.3596 | 0.3948 |
| | lightGCN-M | **0.4381** | **0.5757** | **0.6349** | **0.7137** | **0.4381** | **0.5121** | **0.5311** | **0.5509** |
| | MB-GMN | 0.0384 | 0.1572 | 0.2666 | 0.4212 | 0.0384 | 0.0982 | 0.1321 | 0.1725 |
| | **MB-DP** | **0.4969** | **0.5949** | **0.6553** | **0.7313** | **0.4969** | **0.5482** | **0.5676** | **0.5868** |
| | **Improvement** | **13.43%** | **3.32%** | **3.22%** | **2.47%** | **13.43%** | **7.06%** | **6.88%** | **6.51%** |

performance gap between MB-DP and these baselines demonstrates the advantage of our method in capturing the dynamic preferences of users and items. In contrast, the GNN-based baselines aggregate unique features of behavior perception patterns or user behavior patterns but ignore dynamic preferences.

## C. Model Ablation Study (RQ2)

To evaluate the design submodules of MB-DP, we consider five model variants: **w/o MBeh**, **w/o Meta**, **w/o MGcn**, **w/o Fuse**, and **w/o DI**. These variants respectively remove the use of multiple behaviors in behavior semantic extraction, the low-order personalized semantic module, the high-order connected semantic extraction module, the behavior relationship denoising module, and the dynamic preferences extraction module.

Table III summarizes the results of ablation experiments on different datasets. The following conclusions can be drawn from the experimental results: 1) Using multi-behavioral interaction information is effective in this type of recommendation, as w/o MBeh performs poorly compared to MB-DP. 2) The performance of MB-DP is better than that of w/o Meta, indicating that different behaviors have different meanings for each user and help in user behavior modeling. 3) Graph neural networks can differentiate between user-to-user or item-to-item behavior differences, showing the efficacy of incorporating interaction structure in behavior perception or user behavior patterns. 4) High-order connected semantic extraction by treating all behavior types as a new type reduces behavior noise, and the absence of behavior relationship denoising in w/o Fuse leads to consistently poor performance across various top-N recommendation settings. 5) Dynamic preferences extraction positively impacts recommendation performance as time information helps to obtain users' preference changes for different items and determine their current preference level.

TABLE III. PERFORMANCE COMPARISON OF DIFFERENT VARIANTS

| Metrics | UserBehavior | | IJCAI | |
|---|---|---|---|---|
| | *HR@10* | *NDCG@10* | *HR@10* | *NDCG@10* |
| w/o MBeh | 0.5240 | 0.3906 | 0.5959 | 0.4633 |
| w/o Meta | 0.5534 | 0.4546 | 0.6360 | 0.5452 |
| w/o MGcn | 0.5242 | 0.4277 | 0.6088 | 0.5316 |
| w/o Fuse | 0.4839 | 0.3730 | 0.5856 | 0.4507 |
| w/o DI | 0.5569 | 0.4596 | 0.6445 | 0.5585 |
| MB-DP | 0.5635 | 0.4705 | 0.6553 | 0.5676 |

## D. Hyperparameter Study (RQ3)

We experiment with different parameter settings for MB-DP to evaluate its performance with varying configurations of key hyperparameters: hidden state dimension, low-rank decomposition dimension, and number of layers in the graph neural network. The results are presented in Figure3-Figure5, where the y-axis shows the performance increase ratio compared to the default setting of MB-DP.

The hidden state dimension (*dim*) of MB-DP is varied from 8 to 64, and increasing *dim* improves the recommendation performance, but the improvement became relatively flat when *dim* > 32 for top-N recommendation beyond N = 1. Low-rank decomposition dimension (*rank*) is varied from 4 to 16, and the impact of *rank* on recommendation performance is found to be not significant. We also vary the number of graph neural network layers (*layer*) from 1 to 4 and find that increasing layers decrease the Top-1 recommendation performance due to overfitting, while increasing hit rate for Top-N recommendation with little effect on the recommendation order.

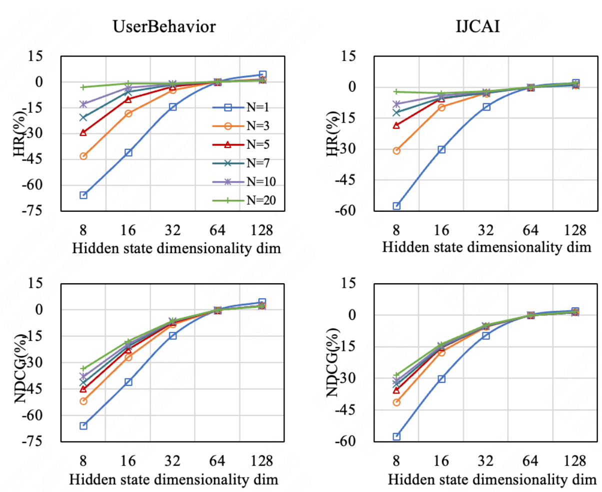


Figure 3. Effect of hidden state dimensionality on MB-DP

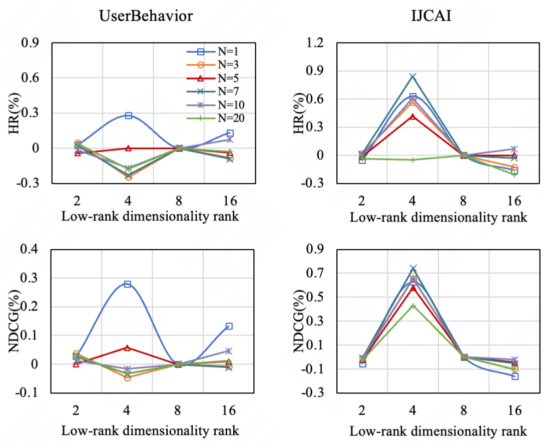


Figure 4. Effect of low-rank dimensionality on MB-DP

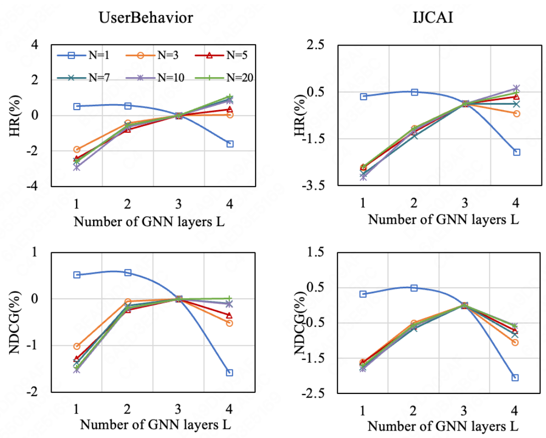


Figure 5. Effect of GNN layers on MB-DP

## VI. Conclusion

In this paper, we propose a multi-behavior recommendation model integrating dynamic preferences (MB-DP), which reduces the impact of noise on recommendation performance when exploiting information from multiple types of interaction behavior and improves recommendation accuracy by utilizing information about users' dynamic interests for items. We conduct extensive experiments on two real-world datasets, and the results show that MB-DP significantly outperforms various state-of-the-art baselines. Furthermore, we find that dynamic preferences contribute to the final recommendation performance. In future work, we will further explore how the connections between behaviors and the diverse semantics of behaviors affect the recommendation quality, and seek general ways to mitigate this impact for better recommendation performance.

## References


[1] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys* 52, 1 (2019), 1-38.

[2] Long Guo, Lifeng Hua, Rongfei Jia, Binqiang Zhao, Xiaobo Wang, and Bin Cui. 2019. 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*, 1984-1992.

[3] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, Lina Yao, Yang Song, and Depeng Jin. 2019. Learning to recommend with multiple cascading behaviors. *IEEE transactions on knowledge and data engineering* 33, 6 (2019), 2588-2601.

[4] Chong Chen, Weizhi Ma, Min Zhang, Zhaowei Wang, Xiuqiang He, Chenyang Wang, Yiqun Liu, and Shaoping Ma. 2021. Graph heterogeneous multi-relational recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 3958-3966.

[5] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Multi-behavior recommendation with graph convolutional networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 659-668.

[6] Lianghao Xia, Yong Xu, Chao Huang, Peng Dai, and Liefeng Bo. 2021. Graph meta network for Multi-behavior recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on research and development in information retrieval*, 757-766.

[7] Yanfu Zhang, Shangqian Gao, Jian Pei, and Heng Huang. 2022. Improving social network embedding via new second-order continuous graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2515-2523.

[8] Jiaxuan You, Yichen Wang, Aditya Pal, Pong Eksombatchai, Chuck Rosenburg, and Jure Leskovec. 2019. Hierarchical temporal convolutional networks for dynamic recommender systems. In *Proceedings of the world wide web Conference*, 2236-2246.

[9] Xiaohan Li, Mengqi Zhang, Shu Wu, Zheng Liu, Liang Wang, and S. Yu Philip. 2020. Dynamic graph collaborative filtering. In *Proceedings of the 2020 IEEE International Conference on Data Mining*, 322-331.

[10] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the Advances in Neural Information Processing Systems* 30, 1025-1035.

[11] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 165-174.

[12] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 639-648.

[13] Wentao Zhang, Ziqi Yin, Zeang Sheng, Yang Li, Wen Ouyang, Xiaosen Li, Yangyu Tao, Zhi Yang, and Bin Cui. 2022. Graph attention multi-layer perceptron. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4560-4570.

[14] Yanping Zheng, Hanzhi Wang, Zhewei Wei, Jiajun Liu, and Sibo Wang. 2022. Instant Graph Neural Networks for Dynamic Graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2605-2615.

[15] Shuyun Gu, Xiao Wang, Chuan Shi, and Ding Xiao. 2022. Self-supervised Graph Neural Networks for Multi-behavior Recommendation. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*. 2052-2058.

[16] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 447-456.

[17] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining*, 197-206.

[18] Jing Lin, Weike Pan, and Zhong Ming. 2020. FISSA: fusing item similarity models with self-attention networks for sequential recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, 130-139.

[19] Zhi Bian, Shaojun Zhou, Hao Fu, Qihong Yang, Zhenqi Sun, Junjie Tang, Guiquan Liu, Kaikui Liu, and Xiaolong Li. 2021. Denoising user-aware memory network for recommendation. In *Proceedings of the 15th ACM Conference on Recommender Systems*, 400-410.

[20] Jinfeng Fang and Xiangfu Meng. 2022. URPI-GRU: An approach of next POI recommendation based on user relationship and preference information. *Knowledge-Based Systems* 256, (2022), 109848.