

Behavior-Contextualized Item Preference Modeling for Multi-Behavior Recommendation

Mingshi Yan
College of Intelligence and
Computing, Tianjin University
China
neo.ms.yan@gmail.com

Fan Liu
School of Computing, National
University of Singapore
Singapore
liufancs@gmail.com

Jing Sun
School of Information and
Communication Engineering, Dalian
Minzu University
China
jingsun@dlnu.edu.cn

Fuming Sun
School of Information and
Communication Engineering, Dalian
Minzu University
China
sunfuming@dlnu.edu.cn

Zhiyong Cheng
School of Computer Science and
Information Engineering, Hefei
University of Technology
China
jason.zy.cheng@gmail.com

Yahong Han*
College of Intelligence and
Computing, Tianjin University
China
yahong@tju.edu.cn

ABSTRACT

In recommender systems, multi-behavior methods have demonstrated their effectiveness in mitigating issues like data sparsity, a common challenge in traditional single-behavior recommendation approaches. These methods typically infer user preferences from various auxiliary behaviors and apply them to the target behavior for recommendations. However, this direct transfer can introduce noise to the target behavior in recommendation, due to variations in user attention across different behaviors. To address this issue, this paper introduces a novel approach, Behavior-Contextualized Item Preference Modeling (BCIPM), for multi-behavior recommendation. Our proposed Behavior-Contextualized Item Preference Network discerns and learns users' specific item preferences within each behavior. It then considers only those preferences relevant to the target behavior for final recommendations, significantly reducing noise from auxiliary behaviors. These auxiliary behaviors are utilized solely for training the network parameters, thereby refining the learning process without compromising the accuracy of the target behavior recommendations. To further enhance the effectiveness of BCIPM, we adopt a strategy of pre-training the initial embeddings. This step is crucial for enriching the item-aware preferences, particularly in scenarios where data related to the target behavior is sparse. Comprehensive experiments conducted on four real-world datasets demonstrate BCIPM's superior performance compared to several leading state-of-the-art models, validating the robustness and efficiency of our proposed approach.

*Y. Han is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).
Conference'17, July 2017, Washington, DC, USA
© 2024 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM... \$15.00
<https://doi.org/XXXXXXXXXXXXXX>

CCS CONCEPTS

• **Information systems** → **Personalization; Recommender systems; Collaborative filtering.**

KEYWORDS

Collaborative Filtering, Multi-behavior Recommendation, Item-aware Preference, Graph Convolutional Networks

ACM Reference Format:

Mingshi Yan, Fan Liu, Jing Sun, Fuming Sun, Zhiyong Cheng, and Yahong Han. 2024. Behavior-Contextualized Item Preference Modeling for Multi-Behavior Recommendation. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 10 pages. <https://doi.org/XXXXXXXXXXXXXX>.

1 INTRODUCTION

With the rapid advancement of the Internet, individuals are inundated with an ever-increasing volume of information and choices. Recommender systems [3, 11] provide an effective way to filter and curate information, thereby saving users time and effort. Collaborative Filtering (CF) [6, 10, 17, 25, 27] plays a pivotal role in recommender systems. CF operates by relying solely on user-item interaction data to learn user and item embeddings, which respectively represent user preferences and item features. It then employs a similarity function to estimate the compatibility between users and items based on their embeddings to provide recommendations. Traditional CF methods, typically focusing on a single behavior such as *purchase*, often encounter challenges with data sparsity, leading to dramatic performance degradation.

In real scenarios, user interactions extend beyond *purchase* to include behaviors like *click* and *cart*, which also provide rich insights into user preferences. To overcome the challenges posed by data sparsity, researchers have introduced Multi-behavior Recommendation (MBR) methods [2, 7, 13], which exploit data from multiple user behaviors to learn user preferences. In these models, while the most desired behavior, typically *purchase*, is considered the target behavior, other user interactions such as *click* and *cart* are utilized as auxiliary behaviors. This approach helps in creating

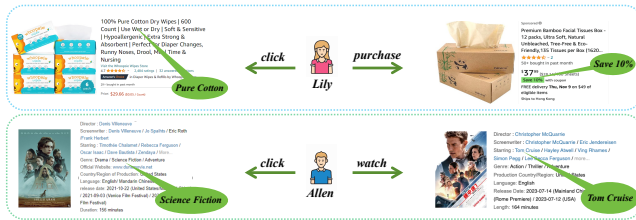


Figure 1: Examples of user behavior influenced by their decision factors.

a more robust and accurate recommendation system by utilizing a broader spectrum of user data.

Recent advancements in MBR methods have marked significant progress, as evidenced by various studies [7, 19, 31, 35]. Initially, early matrix factorization-based approaches [24, 36] utilized shared embeddings to extend matrix factorization across multiple behaviors. Subsequently, with the evolution of Deep Neural Networks (DNN) and Graph Convolutional Networks (GCN), a new wave of methods emerged, combining CF with these advanced technologies. One approach treats multiple behaviors as distinct spaces [2, 28], employing DNN to learn user preferences for each behavior individually. Techniques such as projection are then used to transfer user preferences from auxiliary behaviors to the target behavior space for fusion. Another approach involves utilizing GCN to perform graph convolutions on each behavior separately [8, 13] and then fusing the learned user representations for target behavior recommendation. Despite the technical difference, these methods share a common goal: incorporating user preferences from auxiliary behaviors into the target behavior to enhance user representation.

While current MBR methods have shown effectiveness, they face a significant challenge: the direct information transfer from the auxiliary behaviors to the target behaviors may introduce noise due to the variation of user focus across different behaviors. This issue can be vividly illustrated through real-world scenarios. As depicted in Figure 1, consider the case of Lily, who, when purchasing tissues, is initially drawn to the "pure cotton" for tissues. However, she ultimately chooses a different brand, influenced by a 10% cost-saving. Similarly, Allen, on a movie website, initially clicks on "Dune" due to his interest in the "science fiction" genre, but eventually watches "Mission Impossible" for its star "Tom Cruise". These examples highlight that user interactions across various behaviors are influenced by different aspects of item features. When preferences learned from auxiliary behaviors are merged into the target behavior, they enrich the user's profile but also introduce deviations from their actual preferences in the target behavior. This is attributed to differences in user focus between behaviors. As a result, the partial preference information gleaned from other behaviors can inadvertently introduce noise when applied to the context of the target behavior. It's important to recognize that this variation in user focus is prevalent not only across different behaviors but also within the same behavior, influenced by the specific characteristics of each item. We term these influencing factors as **item-aware preference**. Acknowledging these item-aware preferences allows for a deeper understanding of users' specific points of interest in their interactions. However, current CF-based recommendation methods tend to

overlook these nuanced preferences. Our approach aims to address this gap by considering item-aware preferences, thereby providing a more accurate representation of user interests and enhancing recommendation effectiveness.

In addressing the challenges identified within MBR systems, we propose a novel approach, Behavior-Contextualized Item Preference Modeling (BCIPM), for Multi-Behavior Recommendation. The core of this approach is the Behavior-Contextualized Item Preference Network (BIPN), which extracts item-specific preferences from user-item interactions of various behaviors. BIPN features a three-layer architecture that models the interactions between users and items within specific behaviors. A key feature of BIPN is its strategic use of data from auxiliary behaviors, which is exclusively employed for training the network's parameters. This ensures that auxiliary behaviors do not exert a direct influence on the final recommendation. Our model generates recommendations by aggregating only item-aware preferences relevant to users' target behavior, effectively bypassing the noise introduced by auxiliary behaviors. As random embeddings often lack informative value, we pre-train user and item embeddings via the GCN-based method. This pre-training captures a wider range of collaborative information by using multi-behavior interaction data without distinguishing between behavior types, enriching the initial embeddings with more relevant data. Moreover, to tackle the data sparse issue in the target behavior, which may lead to insufficient item-aware preferences, we introduce a GCN enhancement module. This module is designed to reinforce user preferences specifically within the target behavior, ensuring a more robust and accurate recommendation. We conduct extensive experiments on four different real-world datasets, and our proposed BCIPM significantly surpasses existing methods. The experimental results also demonstrate the effectiveness of BCIPM in reducing the noise from auxiliary behaviors in multi-behavior recommendation systems.

In summary, the main contribution of this work is threefold:

- We introduce the Behavior-Contextualized Item Preference Network, an innovative network designed to distill item-aware preferences from user-item interactions. This network is finely tuned to model the nuanced interactions between individual users and items within specific behaviors.
- We employ GCN to pre-train the initial embeddings with multi-behavior interaction data. This technique allows our model to harness a richer set of collaborative information from these interactions, thereby significantly boosting the learning and accuracy of item-aware preferences.
- Our approach has been rigorously tested through extensive experiments conducted on four real-world datasets. The results consistently demonstrate the superior effectiveness and generality of our model. Compared to state-of-the-art methods, BCIPM shows notable performance improvements, underscoring its practical applicability and robustness in diverse scenarios. Our code is available on GitHub¹ for research purpose.

2 RELATED WORK

Recently, MBR [2, 21, 34, 35] methods have gained significant attention for their effectiveness in mitigating data sparsity issues.

¹<https://github.com/MingshiYan/BIPN>

Based on technological evolution, MBR methods can be broadly classified into three categories: traditional machine learning-based, DNN-based, and GCN-based approaches.

Traditional machine learning-based methods, constituting the initial phase of MBR approaches, primarily focus on enhancing user and item representation learning through data derived from auxiliary behaviors. One such method utilizes matrix factorization [21, 24, 36], designed to refine representation learning by factoring matrices of auxiliary behaviors using shared embeddings. Another strategy involves sampling interaction data from auxiliary behaviors [5, 9, 18] to enrich the primary behavior dataset, thus facilitating a better understanding of user preferences.

DNN-based methods [28, 30, 32] integrate collaborative filtering with deep neural networks. These methods exploit the robust ability of neural networks to process complex, nonlinear relationships, thereby uncovering deeper latent associations between users and items. For instance, Xin et al. [32] combined neural networks and attention mechanisms to model item-based collaborative filtering networks and relation-based collaborative filtering networks separately. Moreover, within the DNN framework, these methods have been combined with other recent technological advancements [2, 7].

GCN-based methods [8, 13, 20, 29] leverage GCNs to model collaborative information between users and items. The strength of GCNs in representing graph-structured data has proven advantageous in this context. A primary method, proposed by Jin et al. [13], involves applying GCNs to learn user preferences within each behavioral graph, followed by an aggregation process. Moreover, GCN-based approaches have incorporated advanced techniques [1, 19, 33, 35]. For example, Luo et al. [1] implemented a non-sampling strategy and developed a relation-aware GCN propagation layer through multi-task learning, embedding both node representations and relationships within the graph.

While these MBR methods vary in their technical approaches, they commonly rely on collaborative information from auxiliary behaviors to enhance user representation learning. Traditional machine learning-based methods derive this information directly from user-item interactions within auxiliary behaviors. In contrast, DNN and GCN-based methods model the overarching auxiliary behavior. However, these approaches tend to overlook the potential noise introduced by auxiliary behaviors. Our proposed method contrasts with these by focusing on enhancing the accuracy of user preference learning through item-aware preference modeling within specific user behaviors. In our approach, auxiliary behavior data is utilized solely for embedding pre-training and network parameter optimization, effectively avoiding the introduction of noise.

3 METHODOLOGY

3.1 Preliminaries

In this section, we will provide a detailed description of the BCIPM model. Before delving into that, we first introduce the problem under investigation. Giving $\mathcal{U} = \{u_1, \dots, u_m, \dots, u_M\}$ and $\mathcal{I} = \{i_1, \dots, i_n, \dots, i_N\}$ as the set of users and items, where M and N denote the total number of users and items, with u_m and i_n representing the m -th user and n -th item, respectively. We introduce the matrices $P \in \mathbb{R}^{M \times d}$ and $Q \in \mathbb{R}^{N \times d}$, serving as the embedding matrices for users and items, where d is the embedding dimension.

The user and item embeddings, e_u^0 and e_i^0 , are retrieved from P and Q based on identifiers from the sets \mathcal{U} and \mathcal{I} , respectively, and serve as the initial embeddings for users and items. In addition, we define $\mathcal{B} = \{b_1, \dots, b_k, \dots, b_K\}$ as the set of behaviors, and b_K is the target behavior. For the initialization of behaviors, we employ one-hot embedding e_b to indicate a specific behavior b .

Our objective is to optimize these user, item, and behavior embeddings to obtain user and item representation from multi-behavior interaction data. Subsequently, the model aims to output a list of recommended items for a target user, focusing specifically on the target behavior.

3.2 Behavior-Contextualized Item Preference Modeling

The motivation for this work stems from the observation that user interactions are often significantly influenced by item-specific characteristics. This insight leads us to conclude that aggregating these item-aware factors across different items in a specific behavior can provide an accurate representation of a user's preferences for that behavior. Inspired by this, we design a network that focuses on learning item-aware preference embeddings within a specific behavior. To enhance the representation of these item-aware preference embeddings, we begin with pre-training on the initial embeddings. This pre-training leverages data from multiple behaviors, allowing us to utilize a broader and more nuanced spectrum of user-item interaction data. This strategy is particularly crucial in addressing the data sparsity issues [16] in recommender systems. Moreover, we recognize that pre-training alone might not suffice in fully capturing the item-aware preference embeddings for the target behavior, which could result in sub-optimal recommendation performance. To mitigate this potential shortfall, we further refine and enhance the user preference embeddings specifically for the target behavior.

As depicted in Figure 2, our model consists of three components: (1) **Embedding Pre-training**: This component is designed to pre-train the user and item embeddings based on the randomly initialized embeddings and collaborative information. (2) **Behavior-Contextualized Item Preference Network**: This is the core component of our model. It focuses on learning behavior-contextualized item preferences which capture the nuanced interplay between users' interactions and item characteristics within specific behaviors. (3) **GCN Enhancement Module**: This module plays a crucial role in enhancing user preferences, particularly targeting the potential issue caused by data sparsity in the target behavior. In the next, we will provide a detailed introduction to these three components.

3.2.1 Embedding Pre-training. Prior to capturing item-aware preference, it is essential to acknowledge that user and item embeddings are typically initialized randomly using specific strategies (e.g., normal distribution or uniform distribution) and inherently lack valuable information. However, imbuing the embeddings with collaborative information is a prerequisite for learning the item-aware preference. Consequently, it becomes necessary to pre-train the embeddings before inputting them into the behavior-contextualized item preference network.

Given its notable effectiveness in representation learning, we employ the GCN technique for embedding pre-training. In this

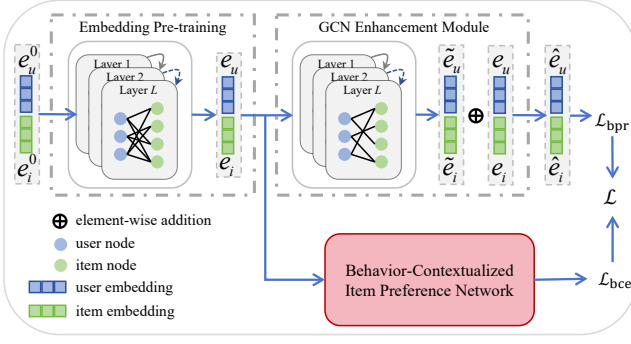


Figure 2: Overview of our proposed BCIPM.

paper, we do not distinguish between different types of behaviors, but instead apply graph convolution to a single unified graph created from interaction data encompassing all behaviors. It is justified by the idea that ignoring behavior types allows for learning fine-grained representations within a broader graph. Additionally, these representations are not influenced by behavioral factors (See Section 4.3.4 for detailed experimental analysis).

Owing to the effectiveness of LightGCN [10], it is utilized to perform graph convolution on the constructed unified graph. Specifically, we construct an interaction matrix $R \in \mathbb{R}^{M \times N}$ with user-item interactions from all behaviors, where $M(N)$ represents the total number of users (items). When observing a user-item interaction in any behavior, the corresponding element r_{ui} in the matrix R is set to 1, otherwise 0. Here, we briefly recap the application of LightGCN on the interaction matrix R ,

$$A = \begin{bmatrix} \mathbf{0} & R \\ R^T & \mathbf{0} \end{bmatrix}, \quad (1)$$

where A is the adjacency matrix of the graph. Next, the information from high-order neighborhoods is aggregated through the graph convolution operation,

$$E^{(l+1)} = (D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) E^{(l)}, \quad (2)$$

where $E^{(l)}$ represents the l -th layer node embedding, when $l = 0$, the $E^{(0)}$ is the initialization of the user and item embeddings, and D denotes the diagonal identity matrix. Finally, we aggregate the layer node embeddings,

$$E = \sum_{l=0}^L \alpha_l E^{(l)}, \quad (3)$$

where α is the weight coefficient, and following LightGCN, we set it as $1/(l+1)$. E represents the embeddings of users and items, serving as inputs for subsequent modules, with its embedding form denoted as e_u for user u and e_i for item i .

3.2.2 Behavior-Contextualized Item Preference Network.

The core of the BIPN, as shown in Figure 3, is dedicated to unveiling users' item-aware preference in a specific behavior. This network is structured with a **Pre-filtering Layer**, an **Item-aware Layer**, and a **Post-filtering Layer**. The network takes a triplet $\langle e_u, e_i, e_b \rangle$ as input, where e_u and e_i denote the pre-trained embeddings of the

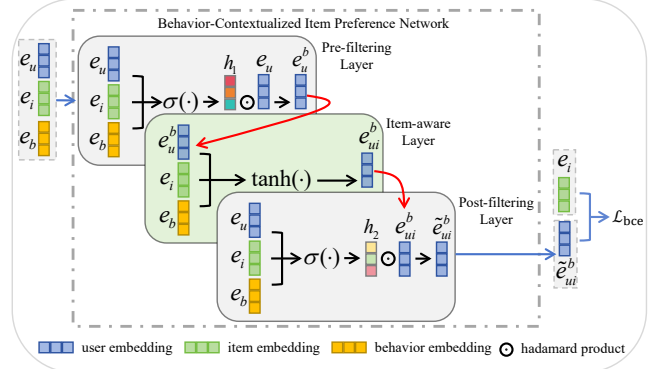


Figure 3: Structure of the Behavior-Contextualized Item Preference Network.

user u and item i , respectively, and e_b represents the embedding of the behavior b .

Pre-filtering Layer. This layer primarily functions to filter the input user embeddings. As discussed in the preceding section, the pre-trained embeddings are not influenced by behavioral factors. Therefore, it becomes imperative to filter user embeddings, retaining only relevant information aligned with the current behavior before learning item-aware preference in a specific behavior. To achieve this, triplet $\langle e_u, e_i, e_b \rangle$ are employed to learn a weight vector $h_1 \in \mathbb{R}^d$. This vector is used to perform element-wise multiplication on user embeddings, facilitating element-level information filtering [26],

$$h_1 = \sigma(W_1(e_u || e_i || e_b) + b_1), \quad (4)$$

$$e_u^b = h_1 \odot e_u, \quad (5)$$

where e_u^b represents the preference information related to item i that user u may possess within behavior b , σ is the sigmoid function, $W_1 \in \mathbb{R}^{d \times (2d+l)}$ and $b_1 \in \mathbb{R}^d$ are trainable parameters, d and l denote the embedding dimensions of the user (item) and behavior, respectively. $||$ means the concatenation operation, \odot represents the Hadamard product.

Item-aware Layer. In this layer, we aim to capture the user's specific preference when interacting with a particular item in a behavior-contextualized setting. It can be formulated as:

$$e_{ui}^b = \tanh(W_2(e_u^b || e_i || e_b) + b_2), \quad (6)$$

where e_{ui}^b represents the item-aware preference of user u for item i in behavior b , $W_2 \in \mathbb{R}^{d \times (2d+l)}$ and $b_2 \in \mathbb{R}^d$ are trainable parameters.

Post-filtering Layer. In the final layer, we apply additional filtering to the learned preferences, similar to the pre-filter layer. This layer is designed to capture better item-aware behavior-specific preference, and it is formulated as:

$$h_2 = \sigma(W_3(e_u || e_i || e_b) + b_3), \quad (7)$$

$$\tilde{e}_{ui}^b = h_2 \odot e_{ui}^b, \quad (8)$$

where \tilde{e}_{ui}^b represents the behavior-contextualized item preference embedding of user u for item i in behavior b .

Intuitively, the higher the correlation between item-aware preferences with the item, the more accurate the learning of item-aware preferences will be. Here, we adopt cross-entropy loss to optimize the network,

$$\mathcal{L}_{bce} = - \sum y \cdot \log \sigma(y') + (1 - y) \cdot \log(1 - \sigma(y')), \quad (9)$$

where y is the ground truth, denoting 1 when the u - i interaction is observed in behavior b_k and 0 otherwise, $y' = (\tilde{\mathbf{e}}_{ui}^{b_k})^\top \mathbf{e}_i$, and $b_k \in \mathcal{B}$.

User's target behavior preference learning. After training the BIPN with the interaction data of all behaviors, we then use the trained network to learn user preferences in the target behavior. Specifically, for a user u , we aggregate the item-aware preferences learned from all interacted items in the target behavior (denoted as \mathcal{I}_{b_K}) to represent the user's specific preference $\tilde{\mathbf{e}}_u^{agg}$ in the target behavior,

$$\tilde{\mathbf{e}}_u^{agg} = \sum_{i \in \mathcal{I}_{b_K}} \tilde{\mathbf{e}}_{ui}^{b_K}, \quad (10)$$

where b_K refers to the target behavior.

3.2.3 GCN Enhancement Module. In the above sections, we learn user preference in the target behavior using the BIPN with the pre-trained embeddings, which are optimized using data of all behaviors. Note that BIPN is a preference-filtering network to identify item-aware preference, it is not a collaborative filtering model for a specific behavior. When the user's interaction records are insufficient in the target behavior, the aggregated user preference by Eq. 10 may fail to accurately reflect the user preferences. To address this issue, we leverage GCN's ability to explore higher-order neighbor information to enhance user preferences. For simplicity, we directly adopt LightGCN. As described in Section 3.2.1, with E , which is the pre-trained embeddings of users and items, as the initialization, to perform graph convolution operation on the target behavior, resulting in node embeddings denoted as E' . We then aggregate E and E' to enhance the representations of users and items in the target behavior,

$$\hat{E} = E \oplus E', \quad (11)$$

where \oplus denotes the element-wise addition, \hat{E} represents users and items embeddings in the target behavior (matrix form), represented in embedding form as $\hat{\mathbf{e}}_u$ and $\hat{\mathbf{e}}_i$. When there is an insufficient record of user interaction in the target behavior, $\hat{\mathbf{e}}_u$ and $\hat{\mathbf{e}}_i$ contribute to enhancing the accuracy of recommendation. We will provide a detailed explanation in the next section.

3.2.4 Prediction & Optimization. In this section, we will introduce the prediction and model optimization of the proposed model.

Prediction. Currently, we have obtained two representations, $\tilde{\mathbf{e}}_u^{agg}$ and $\hat{\mathbf{e}}_u$, for user u in the target behavior. $\tilde{\mathbf{e}}_u^{agg}$ represents the user's specific interests learned from individual interactions. It is characterized by a higher level of specificity and detail. $\hat{\mathbf{e}}_u$ represents similar preferences aggregated from high-order neighbors using GCN, and it tends to be more generalized in nature. When ample user interaction records exist in the target behavior, $\tilde{\mathbf{e}}_u^{agg}$ can more accurately represent user preferences. Conversely, in scenarios where there is a scarcity of user interaction records, the accuracy of $\tilde{\mathbf{e}}_u^{agg}$ cannot be guaranteed. In such cases, $\hat{\mathbf{e}}_u$ can provide

enhanced recommendations by relying on preferences aggregated from higher-order neighbors. Hence, we configure hyperparameter $\lambda = 1/N_u$ to govern the influence of these two representations on recommendation, where N_u is the number of user interactions in the target behavior. A comprehensive analysis is presented in Section 4.4 of the experiment.

Following previous works [4, 13, 34], we adopt the inner product score y_{ui} to evaluate the user's interest in items to provide personalized recommendation,

$$y_{ui} = (1 - \lambda) \cdot (\tilde{\mathbf{e}}_u^{agg})^\top \mathbf{e}_i + \lambda \cdot (\hat{\mathbf{e}}_u)^\top \hat{\mathbf{e}}_i, \quad (12)$$

where $\tilde{\mathbf{e}}_u^{agg}$ represents the user preference aggregated by the target behavior item-aware preferences of user u , \mathbf{e}_i represents the embedding of item i obtained through embedding pre-training. $\hat{\mathbf{e}}_u$ and $\hat{\mathbf{e}}_i$ correspond to the embeddings of user u and item i learned by the GCN enhancement module, respectively.

Optimization. Following the previous CF-based works [4, 8, 10], the Bayesian Personalized Ranking (BPR) loss [22] is adopted for the recommendation performance optimization,

$$\mathcal{L}_{bpr} = - \sum_{(u,i,j) \in O} \ln \sigma(y_{ui} - y_{uj}), \quad (13)$$

where $O = \{(u, i, j) | (u, i) \in \mathcal{R}^+, (u, j) \in \mathcal{R}^-\}$, and \mathcal{R}^+ and \mathcal{R}^- represent the observed and unobserved interactions in the target behavior, respectively.

Finally, we combine the above two objective functions and employ multi-task learning for joint optimization,

$$\mathcal{L} = \beta \cdot \mathcal{L}_{bce} + (1 - \beta) \cdot \mathcal{L}_{bpr} + \gamma \cdot \|\Theta\|_2, \quad (14)$$

where β and γ are hyperparameters, Θ represents all of the parameters in our model, and $\|\cdot\|_2$ denotes the L_2 regularization.

4 EXPERIMENT

4.1 Experiment Settings

4.1.1 Dataset. Four real-world datasets that are used in our experiments:

- **Taobao.**² This dataset is an e-commerce website under the Alibaba Group. It encompasses three distinct user behaviors: *click*, *cart*, and *purchase*, with the *purchase* being the target behavior.
- **Tmall.**³ Similar to Taobao, Tmall is another e-commerce site owned by Alibaba. This dataset includes four types of user behaviors: *click*, *collect*, *cart*, and *purchase*, with the *purchase* being the target behavior.
- **Yelp.**⁴ Yelp is a well-known review website for businesses in the United States. This dataset includes *tips* and *ratings*. Following previous works [29–31], *ratings* (r) are categorized into three behaviors: *dislike* ($r \leq 2$), *neutral* ($2 < r < 4$), and *like* ($r \geq 4$), with *like* being the target behavior. *tips*, *dislike*, and *neutral* are considered as auxiliary behaviors.
- **ML10M.**⁵ MovieLens 10M dataset, which is a publicly available dataset developed by the GroupLens project for movie recommendation. Similar to the Yelp dataset, user ratings in this dataset

²<https://tianchi.aliyun.com/dataset/649>

³<https://tianchi.aliyun.com/dataset/140281>

⁴<https://www.kaggle.com/yelp-dataset/yelp-dataset>

⁵<https://grouplens.org/datasets/movielens/10m/>

are also categorized into three behaviors: *dislike*, *neutral*, and *like*, with *like* being the target behavior.

For the four datasets, to eliminate duplicate data, we follow the previous works [7, 13] to retain only the earliest occurrence of each interaction. The specific statistical information of the four datasets is presented in Table 1.

Table 1: Statistics of four datasets (“# tar. avg.” represents the average number of user interactions in the target behavior).

Dataset	# User	# Item	# Interaction	# tar. avg.	Behavior Type
Taobao	48,749	39,493	2.0×10^6	3	{click, cart, purchase}
Tmall	41,738	11,953	2.3×10^6	4	{click, collect, cart, purchase}
Yelp	19,800	22,734	1.4×10^6	33	{tips, dislike, neutral, like}
ML10M	67,788	8,704	9.9×10^6	72	{dislike, neutral, like}

4.1.2 Evaluation Protocols & Parameter Settings. To evaluate the effectiveness of our model, we employ two widely used evaluation metrics [4, 13, 19], *Hit Ratio (HR@K)* [14] and *Normalized Discounted Cumulative Gain (NDCG@K)* [12], by considering the *Top@K* recommended items. *HR@K* focuses on the number of items that actually interacted with the user in the recommendation list, while *NDCG@K* takes into account the ranking of items in the recommendation list. Besides, we utilize the leave-one-out validation method [7, 8, 13] for model evaluation.

For all methods, we uniformly set the batch size to 1024 and the embedding size to 64 during the training phase. Our model uses the Adam [15] optimizer for optimization. To determine the optimal values for the hyperparameters, including the learning rate, β , and γ , we perform a grid search on the sets $[1e^{-2}, 3e^{-3}, 1e^{-3}, 1e^{-4}]$, $[0.1, 0.3, 0.5, 0.7, 0.9]$ and $[1e^{-2}, 1e^{-3}, 1e^{-4}]$, respectively. For all GCN modules, we uniformly fixed the number of GCN layers at 2. For the method with cross-entropy loss function, we randomly select four negative samples for each positive sample. In addition, to ensure fairness, we also set parameters for the baselines according to the descriptions in their papers and perform a grid search to find the optimal values.

4.1.3 Baselines. To validate its effectiveness, we compare our model against nine baseline methods. Two of these are single-behavior methods, while the remaining seven are multi-behavior methods.

Single-behavior models:

- **MF-BPR** [22]. This is a matrix factorization-based recommendation approach. It computes user-item inner products for recommendations and utilizes the BPR loss function for optimization.
- **LightGCN** [10]. This method offers a simplified GCN-based approach, cutting out unnecessary non-linear designs. This simplification boosts recommendation performance while also reducing model complexity.

Multi-behavior models:

- **R-GCN** [23]. While R-GCN is not designed for recommendation tasks, its applicability to such domains is being demonstrated. The model has offered valuable insights for extracting information from heterogeneous graph networks.
- **NMTR** [7]. NMTR is an approach that combines DNN with CF. It links different behaviors by sequentially propagating interaction scores and uses multi-task learning to optimize the model.

- **MBGCN** [13]. The model learns user preferences within behavior-specific graphs and combines them based on behavior importance. It also uses an item-item graph network to enhance recommendation capabilities by learning item embeddings.
- **S-MBRec** [8]. The model learns node representations within each behavior’s graphs. It then combines user and item features separately and includes a star contrast learning module to explore similarities between different behaviors.
- **CRGCN** [34]. CRGCN presents a method for modeling behavior sequences. It uses a residual module based on GCN for each behavior and facilitates information transfer between modules through embedding propagation, establishing dependency relationships between behaviors.
- **MB-GCGN** [4]. This model simplifies CRGCN by using LightGCN directly to capture user preferences within each behavior and conducting feature transformation before information propagation.
- **PKEF** [19]. This model builds on MB-CGCN by adding a parallel knowledge fusion module and a projection disentangling multi-expert network. These designs aim to address imbalanced data distribution in multiple behaviors.

4.2 Overall Performance

Table 2 showcases the comparative experimental results of our proposed model alongside nine benchmark baseline models, evaluated across four distinct datasets.

From the experimental results of the single-behavior methods, LightGCN consistently surpasses MF-BPR in performance across all four datasets. This superior performance is due to the enhanced representation learning capability inherent in Graph Convolutional Networks (GCNs). LightGCN is adept at assimilating information from high-order neighbors, in contrast to MF-BPR, which is constrained to deriving insights solely from direct interactions.

The multi-behavior models yielded better results than the single-behavior method on Tmall and Taobao, indicating the effectiveness of involving multiple behaviors. However, R-GCN underperforms LightGCN on both Yelp and ML10M. This is because R-GCN incorporates user preferences from auxiliary behaviors by summing them directly without differentiating various behaviors. In NMTR, a sequential modeling approach establishes connections between behaviors by propagating prediction scores backward, leading to superior performance compared to the single-behavior method. CRGCN, also grounded in sequential modeling, facilitates more extensive collaborative information sharing via embeddings, thus outperforming NMTR. MBGCN exhibits a significantly superior performance compared to R-GCN by assigning distinct weights to different behaviors. S-MBREC has developed unique fusion strategies for both users and items, resulting in improved performance compared to MBGCN. Furthermore, both MBGCN and S-MBREC outperform R-GCN when auxiliary behaviors are included, indicating that noise is introduced and effective fusion strategies can help filter out such noise. Both MB-CGCN and PKEF are variations of CRGCN. MB-CGCN simplifies CRGCN while PKEF adds a parallel knowledge fusion module to MB-CGCN, which helps to address

Table 2: Overall performance across four datasets ("*Impr.*" represents the relative performance improvement compared to the suboptimal metric, the bold font indicates the optimal metric, and the underline represents the suboptimal metric).

Dataset	Metric	Single-behavior		Multi-behavior							<i>Impr.</i>	
		MF-BPR	LightGCN	R-GCN	NMTR	MBGCN	S-MBRec	CRGCN	MB-CGCN	PKEF		BCIPM
Tmall	HR@10	0.0230	0.0393	0.0316	0.0517	0.0549	0.0694	0.0840	0.1073	<u>0.1118</u>	0.1414	26.48%
	NDCG@10	0.0124	0.0209	0.0157	0.0250	0.0285	0.0362	0.0442	0.0416	<u>0.0630</u>	0.0741	17.62%
Taobao	HR@10	0.0178	0.0254	0.0283	0.0409	0.0434	0.0571	<u>0.1152</u>	0.0989	0.1097	0.1292	12.15%
	NDCG@10	0.0101	0.0138	0.0148	0.0212	0.0259	0.0331	<u>0.0629</u>	0.0470	0.0627	0.0716	13.83%
Yelp	HR@10	0.0327	0.0400	0.0348	0.0324	0.0356	0.0353	0.0367	0.0355	<u>0.0423</u>	0.0502	18.68%
	NDCG@10	0.0159	0.0202	0.0176	0.0160	0.0183	0.0173	0.0178	0.0164	<u>0.0210</u>	0.0244	16.19%
ML10M	HR@10	0.0585	<u>0.0662</u>	0.0552	0.0451	0.0469	0.0340	0.0502	0.0627	0.0591	0.0810	22.36%
	NDCG@10	0.0274	<u>0.0319</u>	0.0262	0.0202	0.0228	0.0164	0.0239	0.0302	0.0264	0.0392	22.88%

different issues. Both models have demonstrated outstanding performance on the two datasets, but each has specific strengths and weaknesses on different datasets.

After analyzing the experimental results on Yelp and ML10M, we observed an interesting trend: multi-behavior methods do not consistently improve performance and, in some cases, they perform worse than the single-behavior method, LightGCN. This phenomenon can be attributed to two key factors. First, the nature of multi-behavior data in Tmall and Taobao shows a strong correlation among different behaviors, which enables the extraction of more valuable information from auxiliary behaviors. In contrast, the Yelp and ML10M datasets exhibit a weaker correlation among behaviors, resulting in a diminished utility of information derived from auxiliary sources. Second, there is a significant difference in the average number of user interactions within the target behaviors of different datasets. Specifically, Yelp and ML10M have higher interaction numbers, whereas Taobao and Tmall exhibit lower interaction numbers in the target behaviors (as detailed in Table 1). Consequently, integrating auxiliary behaviors can help mitigate data sparsity issues on Taobao and Tmall. However, in Yelp and ML10M scenarios, this integration tends to introduce more noise rather than valuable information. These observations align with our hypothesis that incorporating user preferences from auxiliary behaviors can lead to noise.

Our proposed BCIPM model sets a new benchmark in achieving state-of-the-art performance across all four datasets. These experimental results strongly substantiate its efficacy and versatility. The exceptional performance can be credited to its two-pronged design. The BIPN adeptly discerns user-specific and detailed preferences, while its GCN enhancement module adeptly understands users' general preferences. Furthermore, the incorporation of embedding pre-training is pivotal, ensuring a thorough learning of both detailed and general aspects of user preferences. To summarize, we have reached the following conclusions: (1) The incorporation of multiple behaviors substantially improves the performance of recommendation systems; (2) When integrating multiple behaviors, it is crucial to consider the potential issue of introducing noise; (3) Our model demonstrates robust adaptability and effectiveness across various datasets. In the forthcoming ablation study, we will delve into a detailed examination of the contribution and effectiveness of each individual module.

Table 3: Ablation study of the individual design components ("*w/o. pre.*", "*w/o. inh.*" and "*w/o. net.*" represent methods that without embedding pre-training module, without GCN enhancement module, and without BIPN module, respectively).

Dataset	Metric	<i>w/o. pre.</i>	<i>w/o. inh.</i>	<i>w/o. net.</i>	<i>our</i>
Tmall	HR@10	0.0559	0.0863	0.1043	0.1414
	NDCG@10	0.0277	0.0405	0.0580	0.0741
Taobao	HR@10	0.0346	0.1025	0.1147	0.1292
	NDCG@10	0.0179	0.0560	0.0667	0.0716
Yelp	HR@10	0.0400	0.0343	0.0296	0.0502
	NDCG@10	0.0195	0.0159	0.0132	0.0244
ML10M	HR@10	0.0775	0.0803	0.0710	0.0810
	NDCG@10	0.0372	0.0374	0.0327	0.0392

4.3 Ablation Study

4.3.1 Component Effectiveness Analysis. To evaluate the effectiveness of the individual design components in our proposed method, this section presents ablation experiments for each component. The specific experimental setups are as follows: (1) **Removing the embedding pre-training module (*w/o. pre.*):** In this experiment, the embedding pre-training module is excluded, and randomly initialized embeddings are directly used as the input of the downstream modules. (2) **Removing the GCN enhancement module (*w/o. inh.*):** In this experiment, we exclude the GCN enhancement module and only aggregate the learned item-aware preferences from target behavior for recommendation. (3) **Removing the behavior-contextualized item preference network module (*w/o. net.*):** In this experiment, we remove the BIPN module and only retain the user and item embeddings learned by the GCN enhancement module for recommendation. The experimental results are reported in Table 3.

For the method *w/o. pre.*, there is a remarkable decrease in performance on the Tmall and Taobao datasets. Our analysis suggests that this decline is associated with the number of user interactions in the datasets. Specifically, in the Tmall and Taobao datasets, where there are few user interaction records in the target behavior (see Table 1), leading to challenges in mining sufficient useful information. This result further confirms the importance of embedding pre-training, particularly in situations with limited interaction data. It enables the utilization of auxiliary behaviors to gain more collaborative information. When there are sufficient user interaction

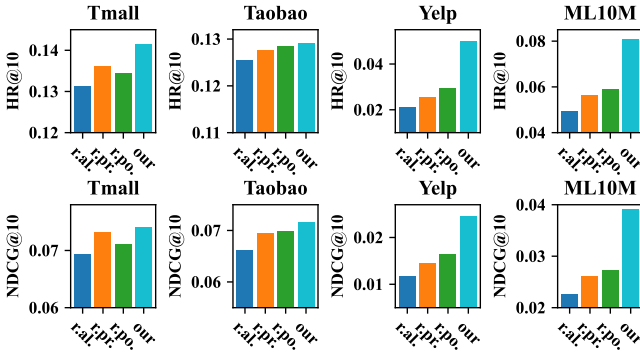


Figure 4: Behavior-contextualized item preference network analysis ("*r.al.*" indicates remove both pre-filtering layer and post-filtering layer, "*r.pr.*" means remove pre-filtering layer, "*r.po.*" means remove post-filtering layer, and "*our*" is our method).

records in the target behavior, the behavior-contextualized item preference network can learn user-specific and detailed preferences, leading to more accurate recommendations. Therefore, in the Yelp and ML10M datasets, *w/o. enh.* performs better than *w/o. net.*. However, when there are fewer user interaction records in the target behavior, the learning capability of the behavior-contextualized item preference network may be insufficient. In the Tmall and Taobao datasets, the GCN enhancement module can improve recommendation quality by aggregating information from high-order neighbors. This demonstrates the superior performance of *w/o. net.*. Our approach (see "*our*") surpasses both *w/o. enh.* and *w/o. net.* in all four datasets, benefiting from the combination of the two types of preferences mentioned above. A detailed discussion will be provided in Section 4.4.

In summary, the experimental results indicate that the removal of any individual module from the model leads to a decline in performance. These results demonstrate the effectiveness and essential contribution of each component within the model's design.

4.3.2 Behavior-Contextualized Item Preference Network Analysis. To verify the effectiveness of the BIPN module, we performed ablation experiments on its components. Specifically: (1) Simultaneously removing the pre-filtering layer and the post-filtering layer (denoted as *r.al.*). (2) Solely removing the pre-filtering layer (denoted as *r.pr.*). (3) Solely removing the post-filtering layer (denoted as *r.po.*). The results of these ablation experiments are illustrated in Figure 4.

The experimental results consistently indicate that in all four datasets, whether removing the pre- or post-filtering layer alone (see *r.pr.* and *r.po.*) or both simultaneously (see *r.al.*), the model's performance significantly decreased. This is because the filtering layers can effectively filter noise based on the embedding triplet $\langle user, item, behavior \rangle$. Compared to the results of removing a single layer of filtering (see *r.pr.* and *r.po.*), the results of simultaneously removing both pre- and post-filtering layers (see *r.al.*) are the worst, indicating that both pre- and post-filtering layers are necessary. In comparison to our method (see *our*), the performance declines after removing the filtering layers (see *r.pr.*, *r.po.* and *r.al.*) in the Tmall

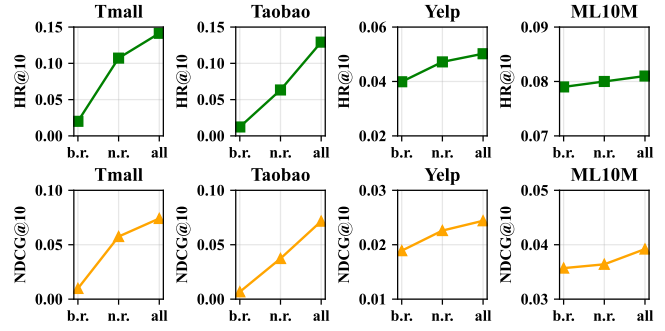


Figure 5: Effectiveness analysis of auxiliary behavior ("*b.r.*" indicates simultaneously eliminate auxiliary behavior from both two modules, "*n.r.*" means exclude auxiliary behaviors solely from the BIPN module, and "*all*" implies keep them all).

and Taobao datasets are less severe than in the Yelp and ML10M datasets. It can be observed that the Yelp and ML10M datasets are more sensitive to changes in the network components due to the allocation of more weight to the BIPN module. This allocation is determined by the quantity of user interactions in the target behavior. On the other hand, the Tmall and Taobao datasets are less susceptible to internal variations as they allocate less weight to this component.

4.3.3 Effectiveness Analysis of Auxiliary Behavior. In our approach, auxiliary behavior is introduced in both the embedding pre-training module and the BIPN module. To verify the effectiveness of incorporating auxiliary behavior, we conducted two experiments: (1) Simultaneously eliminate auxiliary behavior from both the embedding pre-training module and the BIPN module (denoted as *b.r.*). (2) Exclude auxiliary behaviors solely from the BIPN module while retaining them in the embedding pre-training module (denoted as *n.r.*). The experimental results are presented in Figure 5. According to the experimental results, the exclusion of auxiliary behaviors from both the embedding pre-training module and the BIPN module leads to a substantial decrease in performance across all four datasets, thus confirming the necessity of modeling auxiliary behavior data.

Our experiments on Tmall and Taobao showed that removing auxiliary behaviors from both modules simultaneously led to a drastic reduction in the model's predictive capability. This is largely attributed to the sparsity issue of target behavior data within these datasets. The integration of auxiliary behavior data during the embedding pre-training phase allows for the extraction of more collaborative information from an expanded graph. Similarly, incorporating auxiliary behaviors into the BIPN module broadens the training dataset and enhances the optimization of network parameters. In stark contrast, the Yelp and ML10M datasets, which are characterized by a richer set of interaction data in the target behavior, exhibit a more gradual decline in performance following the removal of auxiliary behaviors.

4.3.4 Embedding Pre-training Analysis. In Section 3.2.1, we introduced two distinct methods for embedding pre-training. This

Table 4: Embedding pre-training strategy components ("sep." represents performing GCN individually on each behavior and subsequently aggregating the results, "agg." means performing GCN on the graph constructed from all behavior data without distinguishing behavior types).

Dataset	Metric	sep.	agg.
Tmall	HR@10	0.0948	0.1414
	NDCG@10	0.0505	0.0741
Taobao	HR@10	0.0919	0.1292
	NDCG@10	0.0548	0.0716
Yelp	HR@10	0.0376	0.0502
	NDCG@10	0.0164	0.0244
ML10M	HR@10	0.0722	0.0810
	NDCG@10	0.0332	0.0392

section is dedicated to presenting an experimental comparison between these methods: (1) Performing GCN operations separately in different behaviors and then summing the embeddings obtained from each behavior as the pre-trained embedding (denoted as *sep.*). (2) Employing an aggregated graph constructed from all behavior data without distinguishing behavior types (denoted as *agg.*). The experimental results are reported in Table 4.

The results show that our adopted method *agg.* is significantly superior to the method of conducting GCN operations separately for each behavior and subsequently aggregating (*sep.*). The marked superiority of *agg.* can be attributed to its alignment with the core objective of pre-training, which is to harness a broader scope of collaborative information without being constrained by specific types of behaviors. In *sep.*, there is a potential for information loss during the aggregation process due to the variations in user preferences across different behaviors. Conversely, our approach, which amalgamates data without distinguishing between behavior types, utilizes a graph that integrates interaction data from all behaviors. This integration results in a graph with more interconnected nodes, facilitating the extraction of a more comprehensive array of collaborative information.

4.4 Parameter Analysis

To assess the impact of the hyperparameter λ on the performance of our model, we conducted a series of experiments with λ set at values of 0.3, 0.5, and 0.7. These experiments were aimed at evaluating the performance variations of the BCIPM under different λ settings. The results of these experiments are depicted and analyzed in Figure 6.

In the BCIPM, assigning a larger value to the hyperparameter λ implies an increased emphasis on the role of the GCN enhancement module. In contrast, a smaller λ value highlights the significance of the BIPN module. Our experiments across all four datasets reveal a consistent trend: as λ increases, there is a noticeable decline in BCIPM's performance, particularly in terms of HR@10 and NDCG@10 metrics. This trend underscores the BIPN module as the primary driver of performance improvement. This observation is in perfect harmony with the underlying design of our proposed approach, which prioritizes the capture of item-aware preferences in the target behavior through the BIPN module, relegating the GCN enhancement module to a more ancillary role.

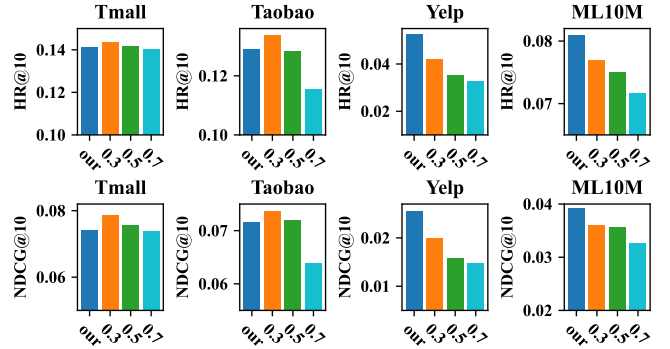


Figure 6: Influence analysis of λ ("our" indicates the configuration parameter used in BCIPM, and the values "0.3", "0.5", and "0.7" represent the corresponding outcomes when λ is set to 0.3, 0.5, and 0.7, respectively).

It is essential to recognize that while the selection of the hyperparameter λ in BCIPM yielded superior performance on Yelp and ML10M, this was not the case for Tmall and Taobao. This variation in outcomes can be ascribed to the differences in user-item interaction frequencies across these datasets. Notably, the Tmall and Taobao datasets exhibit a relatively low average number of interactions between users and items (4 and 3, respectively, as detailed in Table 1). In our approach, we approximately set λ as the inverse of the number of items with which users interact in the target behavior. This heuristic, however, resulted in a sub-optimal allocation of the hyperparameter for numerous users, particularly in contexts where item-aware preferences are insufficient to drive effective recommendations due to limited user-item interactions. In such scenarios, it becomes imperative to assign greater significance to the preferences derived from the GCN enhancement module, ensuring a more robust and effective recommendation.

5 CONCLUSION

In this paper, we revisit the utilization of multi-behavior data in recommendation methods and introduce a novel approach for multi-behavior recommendation. Specifically, our approach employs a two-pronged design, with the behavior-contextualized item preference network designed to learn item-aware preferences and the GCN enhancement module capturing high-order neighbor preferences of users. The combination of these two distinct preferences leads to a significant improvement in recommendation performance. In addition, an embedding pre-training module is adopted to facilitate embedding learning. This module can help users learn item-aware preferences and capture high-order neighbor preferences effectively. Extensive experiments are conducted on four real-world datasets and the experiment results demonstrate the effectiveness of our approach. Moving forward, our research will focus on devising an advanced weight allocation strategy for the GCN enhancement module, to further improve the performance of recommendation model.

ACKNOWLEDGMENTS

This research is supported by the National Natural Science Foundation of China under Grant 62376186, 61932009 and 62272254.

REFERENCES

- [1] 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 35th AAAI Conference on Artificial Intelligence*. AAAI Press, 3958–3966.
- [2] Chong Chen, Min Zhang, Yongfeng Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Efficient Heterogeneous Collaborative Filtering without Negative Sampling for Recommendation. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. AAAI Press, 19–26.
- [3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishu Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Isfir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide and Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 7–10.
- [4] Zhiyong Cheng, Sai Han, Fan Liu, Lei Zhu, Zan Gao, and Yuxin Peng. 2023. Multi-Behavior Recommendation with Cascading Graph Convolution Networks. In *Proceedings of the 32nd ACM Web Conference*. ACM, 1181–1189.
- [5] Jingtao Ding, Guanghui Yu, Xiangnan He, Yuhuan Quan, Yong Li, Tat-Seng Chua, Depeng Jin, and Jiajie Yu. 2018. Improving Implicit Recommender Systems with View Data. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. ijcai.org, 3343–3349.
- [6] Xiao Dong, Lei Zhu, Xuemeng Song, Jingjing Li, and Zhiyong Cheng. 2018. Adaptive Collaborative Similarity Learning for Unsupervised Multi-view Feature Selection. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. ijcai.org, 2064–2070.
- [7] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, Lina Yao, Yang Song, and Depeng Jin. 2021. Learning to Recommend with Multiple Cascading Behaviors. *IEEE Trans. Knowl. Data Eng.* 33, 6 (2021), 2588–2601.
- [8] 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*. ijcai.org, 2052–2058.
- [9] Guibing Guo, Huihui Qiu, Zhenhua Tan, Yuan Liu, Jing Ma, and Xingwei Wang. 2017. Resolving Data Sparsity by Multi-type Auxiliary Implicit Feedback for Recommender Systems. *Knowl. Based Syst.* 138 (2017), 202–207.
- [10] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong 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*. ACM, 639–648.
- [11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web*. ACM, 173–182.
- [12] Kalervo Järvelin and Jaana Kekäläinen. 2000. IR Evaluation Methods for Retrieving Highly Relevant Documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 41–48.
- [13] 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*. ACM, 659–668.
- [14] George Karypis. 2001. Evaluation of Item-Based Top-N Recommendation Algorithms. In *Proceedings of the 10th ACM CIKM International Conference on Information and Knowledge Management*. ACM, 247–254.
- [15] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*. OpenReview.net, 1–15.
- [16] Fan Liu, Huilin Chen, Zhiyong Cheng, Liqiang Nie, and Mohan Kankanhalli. 2023. Semantic-Guided Feature Distillation for Multimodal Recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*. ACM, 6567–6575.
- [17] Fan Liu, Zhiyong Cheng, Lei Zhu, Zan Gao, and Liqiang Nie. 2021. Interest-Aware Message-Passing GCN for Recommendation. In *Proceedings of the 30th Web Conference*. ACM, 1296–1305.
- [18] Babak Loni, Roberto Pagano, Martha A. Larson, and Alan Hanjalic. 2016. Bayesian Personalized Ranking with Multi-Channel User Feedback. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 361–364.
- [19] Chang Meng, Chenhao Zhai, Yu Yang, Hengyu Zhang, and Xiu Li. 2023. Parallel Knowledge Enhancement based Framework for Multi-behavior Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. ACM, 1797–1806.
- [20] Chang Meng, Ziqi Zhao, Wei Guo, Yingxue Zhang, Haolun Wu, Chen Gao, Dong Li, Xiu Li, and Ruiming Tang. 2024. Coarse-to-Fine Knowledge-Enhanced Multi-Interest Learning Framework for Multi-Behavior Recommendation. *ACM Trans. Inf. Syst.* 42, 1 (2024), 30:1–30:27.
- [21] Huihui Qiu, Yun Liu, Guibing Guo, Zhu Sun, Jie Zhang, and Hai Thanh Nguyen. 2018. BPRH: Bayesian Personalized Ranking for Heterogeneous Implicit Feedback. *Inf. Sci.* 453 (2018), 80–98.
- [22] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking From Implicit Feedback. In *Proceedings of the 25th International Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 452–461.
- [23] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *Proceedings of the 15th European Semantic Web Conference*. Springer, 593–607.
- [24] Ajit Paul Singh and Geoffrey J. Gordon. 2008. Relational Learning via Collective Matrix Factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 650–658.
- [25] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. *Adv. Artif. Intell.* 2009 (2009), 421425:1–421425:19.
- [26] Dongjing Wang, Xin Zhang, Yuyu Yin, Dongjin Yu, Guandong Xu, and Shuiguang Deng. 2024. Multi-View Enhanced Graph Attention Network for Session-Based Music Recommendation. *ACM Trans. Inf. Syst.* 42, 1 (2024), 16:1–16:30.
- [27] Yinwei Wei, Wenqi Liu, Fan Liu, Xiang Wang, Liqiang Nie, and Tat-Seng Chua. 2023. Lightgt: A light graph transformer for multimedia recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1508–1517.
- [28] Daqing Wu, Xiao Luo, Zeyu Ma, Chong Chen, Minghua Deng, and Jinwen Ma. 2021. ARGO: Modeling Heterogeneity in E-commerce Recommendation. In *Proceedings of the 34th International Joint Conference on Neural Networks*. IEEE, 1–8.
- [29] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Mengyin Lu, and Liefeng Bo. 2021. Multi-Behavior Enhanced Recommendation with Cross-Interaction Collaborative Relation Modeling. In *Proceedings of the 37th IEEE International Conference on Data Engineering*. IEEE, 1931–1936.
- [30] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Bo Zhang, and Liefeng Bo. 2020. Multiplex Behavioral Relation Learning for Recommendation via Memory Augmented Transformer Network. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2397–2406.
- [31] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Xiyue Zhang, Hongsheng Yang, Jian Pei, and Liefeng Bo. 2021. Knowledge-Enhanced Hierarchical Graph Transformer Network for Multi-Behavior Recommendation. In *Proceedings of the 31th AAAI Conference on Artificial Intelligence*. AAAI Press, 4486–4493.
- [32] Xin Xin, Xiangnan He, Yongfeng Zhang, Yongdong Zhang, and Joemon M. Jose. 2019. Relational Collaborative Filtering: Modeling Multiple Item Relations for Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 125–134.
- [33] Jingcao Xu, Chaokun Wang, Cheng Wu, Yang Song, Kai Zheng, Xiaowei Wang, Changping Wang, Guorui Zhou, and Kun Gai. 2023. Multi-behavior Self-supervised Learning for Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 496–505.
- [34] Mingshi Yan, Zhiyong Cheng, Chen Gao, Jing Sun, Fan Liu, Fuming Sun, and Haojie Li. 2024. Cascading Residual Graph Convolutional Network for Multi-Behavior Recommendation. *ACM Trans. Inf. Syst.* 42, 1 (2024), 10:1–10:26.
- [35] Haoran Yang, Hongxu Chen, Lin Li, Philip S. Yu, and Guandong Xu. 2021. Hyper Meta-Path Contrastive Learning for Multi-Behavior Recommendation. In *Proceedings of the 21st IEEE International Conference on Data Mining*. IEEE, 787–796.
- [36] Zhe Zhao, Zhiyuan Cheng, Lichan Hong, and Ed Hui-hsin Chi. 2015. Improving User Topic Interest Profiles by Behavior Factorization. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 1406–1416.