

Article

A Multi-Behavior and Sequence-Aware Recommendation Method

Dan Yin * and Tianshuo Wang

School of Intelligence Science and Technology, Beijing University of Civil Engineering and Architecture,
Beijing 102616, China

* Correspondence: yindan@bucea.edu.cn

Abstract

This paper proposes a multi-behavior and sequence-aware recommendation method that effectively integrates diverse user–item interaction behaviors and their sequential dependencies to enhance recommendation accuracy. Unlike existing studies that treat different user–item interactions independently, our approach integrates diverse behaviors and their natural sequential dependencies to better capture user preferences and alleviate data sparsity caused by single-behavior modeling. Different from the traditional single-behavior models, our approach constructs a multi-behavior heterogeneous graph and defines multiple meta-path patterns to capture implicit relationships between users and items. By generating subgraph instances, we extract fine-grained interaction patterns and employ a LightGCN with residual connections to learn user representations under different behavioral sequences. Furthermore, an attention mechanism is introduced to fuse features across subgraphs, enabling more expressive preference modeling. Experimental results on two real-world datasets, Taobao and Tmall, demonstrate that our method outperforms state-of-the-art single- and multi-behavior recommendation models, achieving up to 10.0% and 11.1% improvements in HR@10 and NDCG@10 on Taobao and 9.0% and 10.6% on Tmall, respectively. These results confirm the effectiveness of leveraging both multi-behavior information and sequence dependencies in capturing deeper user preferences for more accurate recommendations.

Keywords: multi-behavior; recommendation; meta-path; sequence

1. Introduction

Recommendation systems [1,2] are fundamental techniques for filtering information, providing users with items of potential interest based on their preferences and historical behaviors. Recent advances, including graph attention networks [3,4], contrastive learning (CL) [5,6], and gating mechanisms [7–9], have introduced new paradigms for recommendation, enabling more comprehensive modeling of user preferences. Graph neural network [10–12] based methods learn user and item representations while capturing complex user–item relationships [13–22], whereas graph convolutional network (GCN)-based approaches further exploit higher-order interactions to refine these representations [23–26]. Research based on behavior clustering [27–31] partitions users or items with similar behavioral patterns into groups via clustering algorithms, and then generates recommendations according to the common preferences of each group.

In practice, user behaviors are inherently diverse rather than singular. On e-commerce platforms, users typically exhibit multiple behaviors, such as viewing, adding to cart, and



Academic Editor: Domenico Rosaci

Received: 14 January 2026

Revised: 2 February 2026

Accepted: 3 February 2026

Published: 5 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

purchasing items [16,18,19,23,32–36]. Purchases are treated as the target behavior, whereas auxiliary behaviors provide complementary contextual information that mitigates data sparsity. By leveraging these signals, models can more precisely capture user preferences and thus deliver improved personalized recommendations.

In e-commerce scenarios, user–item interactions exhibit substantial complexity and diversity. To address this, we represent user behaviors on online shopping platforms using a multi-behavior heterogeneous graph. Compared with conventional heterogeneous graphs, this representation provides an enhanced capacity for modeling the multifaceted relationships between users and items. In such a graph, multiple edge types link user and item nodes, with each edge type encoding a distinct interaction behavior. By leveraging the expressive power of this structure, the model can capture richer behavioral information and more accurately characterize user preferences across different semantic levels.

Figure 1 presents a multi-behavior heterogeneous graph constructed from user–item interactions on an e-commerce platform. The graph consists of three types of nodes: users (U), items (I), and item categories (T), as well as four types of edges: buy (a), view (b), add-to-cart (c), and item category (d). For illustration, u_1 and u_2 denote two users, while t_1 , t_2 , and t_3 represent item categories. Item i_1 belongs to item category t_1 , items i_2 and i_3 belong to t_2 , and items i_4 and i_5 belong to t_3 .

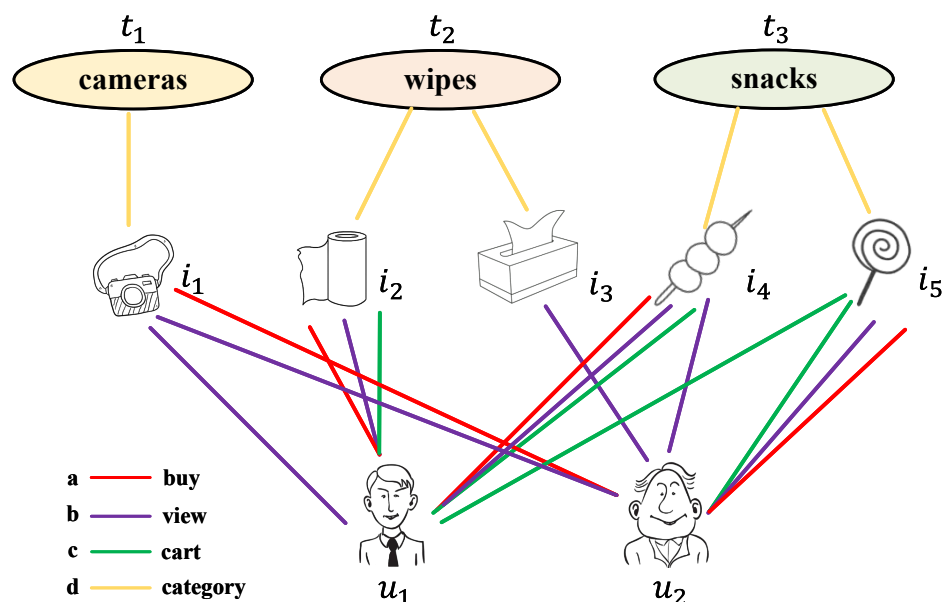


Figure 1. An example of a multi-behavior graph on an e-commerce platform.

Multiple interactions exist between users and items. For example, one user views, adds to the cart, and buys i_2 , whereas user u_2 only views i_3 . Since both i_2 and i_3 belong to category t_2 , the interactions of the two users with items belonging to the same category indicate a potential alignment of their preferences.

User behaviors exhibit sequential characteristics, with different behaviors corresponding to preference signals of varying intensities: in e-commerce scenarios, the behavioral sequence of browsing (weak preference) → adding to cart (strong tendency) → purchasing (explicit preference) clearly reflects the progressive process of user interests. Modeling both the diversity of behaviors and their sequential dependencies can comprehensively depict user interests and optimize recommendation performance. However, existing research still has three key limitations that need further exploration:

1. **Causes of user behaviors:** : Most existing models focus solely on users' personal preferences, ignoring the underlying driving effects of social influence (behaviors of other users) and item categories on user interactions.
2. **Relationship between auxiliary and target behaviors:** In multi-behavior data, there is a hierarchical correlation between auxiliary behaviors (view and add-to-cart) and target behaviors (purchase). However, existing models fail to fully utilize this relationship, making it difficult to accurately capture users' purchase intentions.
3. **Neglect of behavioral sequence:** User behaviors naturally follow a sequential logic such as "view $\rightarrow$ add-to-cart $\rightarrow$ purchase", which reflects the decision-making process and progressive preferences. Yet, existing models do not effectively leverage such sequential dependencies to improve recommendation performance.

To address the identified limitations, we propose a multi-behavior and sequence-aware recommendation framework. Firstly, to capture the underlying causes of user behaviors, we construct a multi-behavior heterogeneous graph based on user–item interactions and item categories, define multiple meta-path patterns, and further divide the graph into subgraphs that embed external factors such as peer interactions and item categories. Secondly, to exploit the relationships between auxiliary and target behaviors, our method leverages multiple user interaction behaviors to learn user and item representations, thereby uncovering latent preferences with higher accuracy. Thirdly, to account for behavioral sequence relationships, the framework explicitly models the hierarchical and sequential dependencies among different behaviors, enabling more precise characterization of user preferences. Through this design, our approach overcomes the shortcomings of existing methods and improves both the accuracy and robustness of personalized recommendations.

In summary, the main contributions of this paper are as follows:

1. We define multiple meta-path patterns and construct corresponding subgraphs. Based on the multiple interactive behaviors between users and items, the model learns latent information among users, items, and item categories from a multi-behavior perspective. This approach alleviates the problem of data sparsity in single-behavior learning and captures deeper-level preference signals across different node types.
2. We integrate the sequential order of different behaviors with multiple meta-path patterns to model user preferences, thereby alleviating data sparsity while incorporating the influence of behavioral sequences [37].
3. We apply residual connections in LightGCN to retain the upstream behavioral features during downstream learning, integrate the latent sequential dependencies into the meta-path sub-learning process, and aggregate the user–item features across various subgraphs via an attention mechanism. A specific normalization method is adopted to mitigate the impact of noise, and the learned node representations of upstream behaviors are used as the input for downstream behavior learning to capture the latent sequential correlations among behaviors.
4. We conduct extensive experiments on historical behavior datasets from Taobao and Tmall. In Top-N recommendation tasks, compared with the baseline model which has the best performance in terms of recent research, our method achieves up to 10.0% and 11.1% improvements in HR@10 and NDCG@10 on Taobao and 9.0% and 10.6% on Tmall, respectively.

The remainder of this paper is organized as follows: First, the Related Work Section reviews the research status in the field of multi-behavior recommendation and clarifies the limitations of existing methods. Second, the Preliminary Section introduces the basic concepts, including the definition of multi-behavior heterogeneous graphs and the concept of multi-meta-path patterns. Third, the Method Section elaborates on the overall architecture of the proposed multi-behavior and sequence-aware recommendation model and the

implementation details of each core module. Fourth, extensive experiments are conducted to verify the effectiveness of the model, including performance comparison, ablation study, parameter analysis, etc. Finally, the Conclusion Section summarizes the entire work and looks forward to future research directions.

2. Related Work

The purpose of multi-behavior recommendation systems is to overcome the limitations of traditional approaches in handling diverse user interests and to mitigate the data sparsity problem inherent in single-behavior learning. Recent studies on multi-behavior recommendation can be broadly categorized into two groups: models that ignore the sequential relationships between behaviors and models that explicitly incorporate such sequential dependencies.

Current multi-behavior recommendation models mainly focused on multiple types of user–item interactions without considering sequential associations between behaviors, treating different behaviors as independent events for modeling. CMF [38] improves prediction accuracy by leveraging relational correlations through joint multi-matrix factorization and parameter sharing. MF-BPR [39] adopts multi-relational modeling, taking user–item interactions as the target relationship and social relationships as auxiliary relationships to solve the cold-start problem. MFBPR [32] regards different interaction methods as “channels”, classifies users into levels based on behavior intensity, and selects positive samples accordingly. BPRH [40] adjusts the sampling frequency according to the correlation between auxiliary and target behaviors to reduce noise interference. CVID [41] combines variational inference with causal graph networks to decouple confounders from behavioral relationships and generate deconfounded user–item representations for enhanced recommendation. DMR [42] combines disentangled representation and denoising modules to separate behavior-specific and shared features, and suppresses noise interference in multi-behavior data via contrastive learning. MBR [43] models multi-behavior recommendation via causal inference, eliminating spurious correlations from unobserved confounders through counterfactual reasoning to boost the prediction accuracy of target behaviors.

Since matrix factorization and sampling strategies struggle to mine deep interaction information, graph neural network-based methods have been proposed successively: MGNN [44] learns shared and behavior-specific embeddings for users and items from the perspective of link prediction in multi-channel networks. GHCF [23] captures high-order interaction relationships through multi-layer graph convolutional networks and fuses user features from multiple behaviors to alleviate data sparsity. MATN [18] models complex dependencies between different interactive behaviors using Transformers and multi-head attention mechanisms. BMGCN [45] designs a behavior-fused graph convolutional network, which dynamically assigns behavior weights and jointly propagates multi-behavior embeddings into a unified collaborative filtering vector. HMGCR [46] integrates embedding representations of multi-behavior patterns based on hyper-meta-graphs and contrastive learning. HyperTEM-MBR [47] models high-order multi-behavior relationships with hyper-graph neural networks and captures the long-term dependencies and dynamic evolution of cross-type behaviors by integrating temporal Transformers. VCGAE [48] designs a variational autoencoder to uniformly encode multi-behavior heterogeneous graphs and generate robust user preference representations via joint decoding.

Currently, only a few studies have focused on behavioral sequence information, and researchers have gradually realized the importance of behavioral order for user preference modeling. NMTR [33] models independent interaction functions for different behavior types through a shared embedding layer and multi-task learning framework. CRGCN [49] combines multi-task learning with graph convolutional networks to learn potential de-

dependencies in the order of behavioral interactions. MBCGCN [37] uses LightGCN [16] to capture sequential relationships among multiple behaviors and integrates upstream behavior embeddings into downstream behavior learning.

Although existing multi-behavior recommendation methods have made certain progress in fusing multi-behavior information or capturing behavioral sequential relationships, they still have notable limitations. Most models only conduct superficial mining of multi-behavior correlations and fail to fully leverage diverse behavioral correlation patterns (e.g., the semantic correlations implied by meta-paths) to construct a systematic correlation modeling framework. Though some models attempt to incorporate behavioral sequential information, they cannot effectively integrate sequential relationships with the deep semantic correlations among multi-behaviors, leading to an incomplete and inaccurate characterization of user preferences. These limitations prevent existing methods from fully exploiting the value of multi-behavior data and meeting the demand for accurate recommendations in complex scenarios. Therefore, developing a method that can integrate diverse meta-path patterns with behavioral sequential relationships and accurately capture the latent dependencies across behaviors and patterns has become an urgent key issue to be addressed in the field of multi-behavior recommendations.

The method proposed in this paper leverages multiple user behaviors by defining diverse meta-path patterns and incorporating sequential relationships among behaviors to capture the latent dependencies across behaviors and patterns. An attention mechanism is then employed to integrate these dependencies, enabling more comprehensive modeling of user-item relationships and yielding more accurate recommendations.

3. Preliminaries

We use u , v , and t to denote a specific user, item, and item type, and U , V , and T to denote the sets of users, items, and item types. The set of interaction matrices between users and items or between items and item types under behavior K is $\mathcal{M} = \{M_1, M_2, \dots, M_K\}$, where if k represents the user-item relationship, then $M_k = [m_{(k)uv}]_{|U| \times |V| \times |T|} \in \{0, 1\}$ denotes the interaction matrix between users and items under behavior k . If k represents the item and item type relationship, M_k represents the adjacency matrix instantiated by item type, which the item belongs to. The interactions between users and items as well as the relationships between items and item types can be regarded as a multi-behavior heterogeneous graph $\mathcal{G} = (\mathcal{O}, \mathcal{E}, \mathcal{M})$, where $\mathcal{O} = U \cup V \cup T$, and $\mathcal{E} = \cup_{k=1}^K \mathcal{E}_k$ represents the set of edges of K types. To represent its sequential structure, we set $k \in \{1, 2, \dots, K\}$, and denote the buy behavior, i.e., the target behavior, as K . A higher value of k signifies a stronger representation of user preferences.

We define the multiple meta-path pattern as $\mathcal{P} = o_1 \xrightarrow{R_1} o_2 \xrightarrow{R_2} \dots \xrightarrow{R_{n-1}} o_n$, which is a set of paths containing multiple node types and edge types, where $o_i \in \mathcal{O}$, $i \in \{1, 2, \dots, n\}$ represents different node types, $R_j \subseteq \mathcal{R}$, $j \in \{1, 2, \dots, n-1\}$ represents the set of edge types, and $|R_j| \geq 1$. In multiple meta-path patterns, there is at least one type of edge between two adjacent nodes. $|\mathcal{P}| = n$ is the length of the multiple meta-paths.

Traditional meta-path research [46,50,51] mostly focuses on mining single semantic associations. For example, it only captures the social influence between users through the $U \rightarrow I \rightarrow U$ path or only analyzes users' preferences for item types through the $U \rightarrow I \rightarrow T$ path. The limitation of such a single path is that it cannot take into account multi-dimensional association information, making it difficult to comprehensively depict users' complex interests. The core principle of the **multi-meta-path pattern**, just like Figure 2, is to deconstruct the user-item interaction graph from multiple dimensions by designing multiple meta-paths with different semantics: on the one hand, through paths such as $U \xrightarrow{R_1} I \xrightarrow{R_2} U$, it explores the behavioral transmission effect between users and restores the

shaping process of social interaction on user preferences; on the other hand, through paths such as $U \xrightarrow{R_1} I \xrightarrow{R_2} U$, it associates item categories with user behaviors and captures users' potential interests in specific categories.

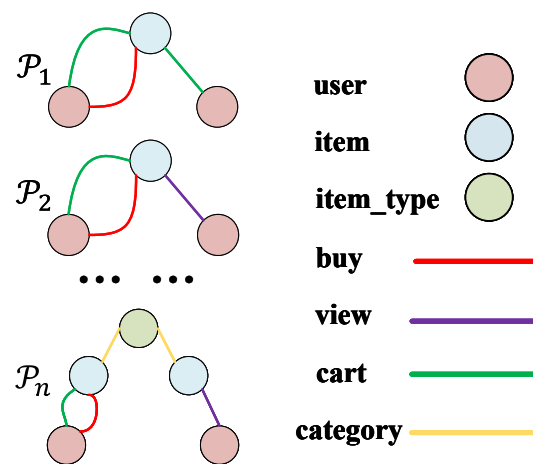


Figure 2. Three multiple meta-path pattern examples.

4. Method Section

We propose a multi-behavior and sequence-aware recommendation model that integrates multiple meta-path patterns with the sequential relationships among diverse user behaviors. Figure 3 presents the overall architecture and technical workflow of the model, which is composed of four main components: (I) Initialize node embeddings and learn unified behavior embeddings. (II) Employ LightGCN to learn the embedding representations e^k of different behaviors in a specific order. (III) According to the defined multiple meta-path patterns, divide the multi-heterogeneous graph into different subgraphs and obtain the embedding representations e^{P_n} under different patterns. (IV) Utilize an attention mechanism [52] to integrate the outputs of steps (I)–(III), optimize the model parameters using BPR loss, and evaluate the recommendation performance.

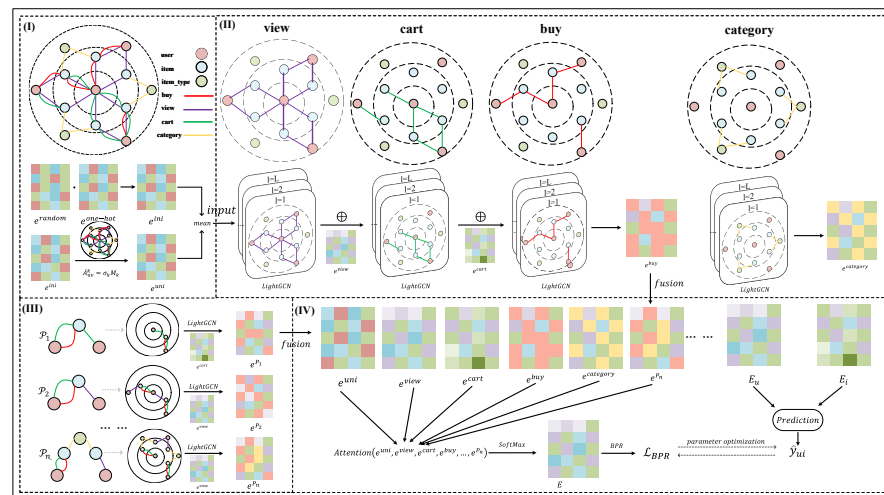


Figure 3. Illustration of the proposed framework. We take the initialized content from Module (I) as the input and feed it into the Behavior Sequence Embedding Module (II), yielding the embedding representations under edges of different types. Meanwhile, we obtain the representations of the target edge via the multi-layer LightGCN, and feed these representations into the designed multi-meta-path pattern (III) for in-depth learning. Subsequently, Module (IV) fuses the information derived from Modules (I–III) by virtue of the attention mechanism to generate the final user and item embeddings. Finally, we optimize the model with a loss function to derive the final vector representations, based on which the Top-k scores are calculated.

4.1. Obtain Initial Embeddings

We first apply one-hot encoding to the IDs of different node types. Then, we combine these encodings with random initialization to generate the initial embedding representations of nodes.

$$e_u = e_u^{\text{random}} \cdot e_u^{\text{one-hot}}, e_v = e_v^{\text{random}} \cdot e_v^{\text{one-hot}}, e_t = e_t^{\text{random}} \cdot e_t^{\text{one-hot}} \quad (1)$$

where $e_u \in \mathbb{R}^{m \times d}$, $e_v \in \mathbb{R}^{n \times d}$, and $e_t \in \mathbb{R}^{s \times d}$ denote the initial embeddings of u , v , and t nodes, respectively, which are constructed by fusing randomly initialized embeddings with one-hot encoded vectors. Herein, m , n , and s represent the cardinality of u , v , and t node sets, while d denotes the embedding dimension. Specifically, e_u^{random} , e_v^{random} , and e_t^{random} refer to the embeddings randomly initialized via the Xavier [53] initialization strategy, and $e_u^{\text{one-hot}}$, $e_v^{\text{one-hot}}$, and $e_t^{\text{one-hot}}$ stand for the one-hot vectors derived from the ID encoding of the three node types. The embedding e^{ini} is acquired by concatenating the initial embeddings of the three node types.

$$e^{\text{ini}} = \text{CONCAT}(e_u, e_v, e_t) \quad (2)$$

4.2. Unified Behavior Embedding Acquisition

LightGCN has been shown to capture user–item interactions more effectively and is particularly well suited for collaborative filtering, thereby enhancing recommendation performance. Therefore, we adopt LightGCN instead of traditional GCN to learn node features. However, the aforementioned initialization methods may yield insufficient information in the initial node embeddings, which can negatively affect subsequent feature learning. To address this, we treat different user–item interactions in the graph as homogeneous edges while assigning distinct weights to different interaction types to derive unified behavior embeddings. The weighted adjacency matrix is then constructed as follows:

$$\hat{A}_{uv}^k = \sigma_k M_k \quad (3)$$

where $\hat{\mathbf{A}}_{uv}^{(k)}$ represents the weighted adjacency matrix of user–item interactions under behavior k , and σ_k represents the learnable weight coefficient for behavior k . The initial parameter σ_k is obtained by random initialization, and a reasonable parameter value is derived through model training. As this is a parameter for the model’s adaptive learning and is randomly initialized, the initial setting of this parameter has virtually no impact on the model’s results. By combining the weighted adjacency matrices under different behaviors k with their respective weight coefficients, we obtain the overall adjacency matrix that encompasses all interaction types and their corresponding weights.

$$\hat{\mathbf{A}}_{uv}^{uni} = \sum_{k=1}^K \hat{\mathbf{A}}_{uv}^k \quad (4)$$

where $\hat{\mathbf{A}}_{uv}^{uni}$ represents the adjacency matrix encompassing all interaction types and their weight coefficients. By combining $\hat{\mathbf{A}}_{uv}^{uni}$ with the initial embedding $\mathbf{e}_i^{ini}$, we employ LightGCN to derive the unified behavior embedding.

$$\mathbf{e}_u^{(l+1)} = \sum_{i \in \mathcal{N}_u} \frac{\hat{\mathbf{A}}_{ui}^{uni}}{\sqrt{\sum_{v \in \mathcal{N}_u} \hat{\mathbf{A}}_{uv}^{uni} \sum_{v \in \mathcal{N}_i} \hat{\mathbf{A}}_{iv}^{uni}}} \mathbf{e}_i^{(l)} \quad (5)$$

where $\mathbf{e}_u^{(l+1)}$ represents the representation of u at the $l + 1$ layer. Similarly, we obtain representations of v and t at the $l + 1$ layer, denoted as $\mathbf{e}_{v(uv)}^{(l+1)}$. When $l = 0$, $\mathbf{e}_i^{(0)} = \mathbf{e}_i^{ini}$. $\mathcal{N}_u$ represents the set of neighbors of u . $\frac{\hat{\mathbf{A}}_{ui}^{uni}}{\sqrt{\sum_{v \in \mathcal{N}_u} \hat{\mathbf{A}}_{uv}^{uni} \sum_{v \in \mathcal{N}_i} \hat{\mathbf{A}}_{iv}^{uni}}}$ represents the core normalization operation in LightGCN. Based on the user–item interaction matrix, we get $\mathbf{e}_{v(uv)}^{(l+1)}$. From the adjacency matrix of items and item types in the multi-behavior heterogeneous graph, we obtain the item embedding matrix related to item types, denoted as $\mathbf{e}_{v(vt)}^{(l+1)}$. Then, we perform an averaging operation to get the item embedding $\mathbf{e}_v^{(l+1)} = \frac{1}{2} (\mathbf{e}_{v(uv)}^{(l+1)} + \mathbf{e}_{v(vt)}^{(l+1)})$. Finally, by concatenating the feature matrices of the three types of nodes, we obtain the node feature matrix under unified behavior, denoted as $\mathbf{e}^{uni}$.

4.3. Behavior Sequence Embedding

Considering that different behaviors have sequential relationships, for example, users will first view an item before purchasing it, and after viewing the item, they may add it to the shopping cart. Therefore, user behaviors form a sequential chain from viewing to adding to the cart and finally to purchasing. According to MBCGCN and others, learning the sequential chain between different behaviors can achieve better recommendations. We adopt this method and improve it by using LightGCN with residual connections. By learning the sequential correlation between different behaviors, we can capture the correlation of user–item interactions between behaviors.

$$\mathbf{e}_{u,0}^{k+1} = \mathbf{e}_{u,0}^k + \text{LightGCN}(M_{k+1}, \mathbf{e}_{u,0}^k) \quad (6)$$

For capturing the sequential correlation between different behaviors of users and items, we use the LightGCN method shown in Equation (6), along with residual connections to obtain the embedding $\mathbf{e}_{u,0}^k$ of user u under behavior k . Similarly, based on the embedding of item v , we use this formula to get the feature representation $\mathbf{e}_{v,0}^k$ of item v under behavior k .

As shown in Equation (6), when learning the embedding of a user under a specific behavior sequence relationship, the adjacency matrix and feature matrix of the user’s previous behavior are used as inputs for the next behavior. In this sequential feature learning, the latter behavior has stronger signals and more accurate user preferences than

the former behavior. Using LightGCN to take the features of the former behavior as inputs for learning the features of the latter behavior can better capture user behavior preferences and learn the potential interests between different user behaviors. After obtaining the representations $\mathbf{e}_{u,o}^k$ and $\mathbf{e}_{v,o}^k$ for users and items, they are fed into multiple meta-path pattern learning.

4.4. Multiple Meta-Path Pattern Learning

To explore the interaction potential between users and items, as well as the correlation between interactions and item types, we split the graph structure into multiple subgraphs using the multi-meta-path patterns shown in Figure 2. These patterns focus on capturing cross-user interaction influences (e.g., the transfer of user preferences across users in the U-I-U path and the extension of this influence on the item type dimension via the U-I-T-I-U path). Selected based on practical experience, they integrate user interaction and item type information to depict latent user preferences. Finally, the quantitative expression of meta-path instances is realized through adjacency matrix calculation:

$$M_{P_{Kk,k}} = \text{sign}((M_K + M_k)) \cdot M_k^\top \quad (7)$$

Equation (7) presents the algorithm for the model in the context of multi-meta-path patterns, and the sign function sets all numbers greater than 0 in the matrix to 1. Here, the $(\cdot)$ does not represent element-wise multiplication but stands for Boolean matrix multiplication, which is used to determine whether there is an edge connection between two nodes. After obtaining this adjacency matrix, we combine it with the user and item representations obtained in Section 3 to acquire deeper-level user interaction information from multiple meta-path patterns and capture potential preferences.

$$e_{u,P_{Kk,k}}^{l+1} = \frac{1}{l+1} \text{LightGCN}(M_{P_{Kk,k}}, e_{u,o}^k) \quad (8)$$

where $\mathbf{e}_{u,P_{Kk,k}}^{(l+1)}$ represents the embedding of user u under the multiple meta-path pattern $P_{Kk,k}$ at layer $l+1$. Considering that LightGCN may introduce some noise when acquiring higher-order information during propagation, the weight for layer $l+1$ is set to $\frac{1}{l+1}$ to properly handle the noise in higher-order information while obtaining higher-order potential content. The content in Section 5.3 can demonstrate the feasibility of this method. After obtaining the embeddings under the multiple meta-path patterns, we integrate the information modules presented in Sections 4.1–4.4. To capture the potential information between users and items, we dynamically assign weights to the embeddings from different aspects or behaviors, enhancing the model's expressive power and recommendation performance by using attention for fusion.

4.5. Attention Fusion

After the above four steps, we obtain the node embedding representation $\mathbf{e}^{\text{ini}}$ initialized by random initialization combined with one-hot encoding, the unified embedding representation $\mathbf{e}^{\text{uni}}$ under the unified graph, the embedding representation $\mathbf{e}_{u,o}^k$ learned under the behavior sequence relationship, and the embedding representation $\mathbf{e}_{u,p}$ learned under the multiple meta-path patterns. By integrating these four aspects and using an attention mechanism for fusion, we get the final user embedding representation $\mathbf{E}_u$:

$$E_u = \text{Attention}\left(a^\top W \left[e^{\text{ini}}, e^{\text{uni}}, e_{u,o}^k, e_{u,p} \right] \right) \quad (9)$$

Here, $\mathbf{a}^\top$ represents the learnable attention vector, and $\mathbf{W}$ is the learnable parameter matrix. Using this formula, we can obtain the final item embedding representation $\mathbf{E}_v$ by fusing the embeddings calculated in steps 1, 2, 3, and 4 related to item v .

4.6. Prediction

After obtaining the user and item embeddings $\mathbf{E}_u$ and $\mathbf{E}_v$, we calculate the similarity between $\mathbf{E}_u$ and $\mathbf{E}_v$ using the following formula to represent the similarity score between user u and item v . This score indicates user u 's preference for item v after model training.

$$\hat{y}_{ui} = \mathbf{E}_u^\top \mathbf{E}_i \quad (10)$$

$\hat{y}_{ui}$ represents the final learned ranking score. The higher the score, the more interested user u is in item i , and the higher the probability that user u will perform the target behavior of purchasing item i .

4.7. Training

Following the majority of multi-behavior tasks [37,44,49,54], we adopt a pairwise learning strategy for model optimization and use the BPR loss with L_2 regularization as the model loss function. In the pairwise learning of this loss, the observed samples are assigned higher scores than the unobserved samples. As the model continues to train, the score difference between the observed and unobserved samples gradually increases.

$$\mathcal{L} = \sum_{(u,i,j) \in O} -\ln \sigma(y_{ui} - y_{uj}) + \lambda \|\Theta\|^2 \quad (11)$$

where $O = \{(u, i, j) \mid (u, i) \in \mathcal{R}^+, (u, j) \in \mathcal{R}^-\}$ represents the set of positive and negative sample pairs, where $(u, i) \in \mathcal{R}^+$ is a positive sample pair and $(u, j) \in \mathcal{R}^-$ is a negative sample pair. $\mathcal{R}^+$ denotes the set of observable interactions in the target behavior, and $\mathcal{R}^-$ denotes the set of unobservable interactions in the target behavior. $\sigma(\cdot)$ is the activation function used in the BPR loss, and here we use the Sigmoid function as the activation function. λ is the weight coefficient controlling L_2 regularization, and Θ represents all the trainable parameters in the model.

5. Experiments

This section presents a comprehensive experimental evaluation of the proposed model to validate its effectiveness. We outline the experimental settings including the datasets, evaluation metrics, baseline models, and parameter configurations. We then report the main performance comparison results, followed by ablation studies to verify the contribution of each core module. Finally, we conduct parameter sensitivity analysis and additional experiments on multi-behavior combinations and behavioral sequences to further validate the rationality of our design.

5.1. Experimental Settings

5.1.1. Experiment Environment

The experiments were conducted using an Intel 8362 CPU and 48 GB of memory, with a NVIDIA GeForce RTX 3090 GPU (24 GB) to ensure efficient data processing capabilities. The software environment included PyTorch version 1.11.0, CUDA version 11.3 for accelerated model training, and Python version 3.8.

5.1.2. Datasets

Both the Taobao and Tmall datasets cover three core user–item interaction behaviors—browsing (Views), adding to cart (Carts), and purchasing (Buys)—which fully reflect the

progressive decision-making process of online shoppers from browsing to placing an order. To integrate item category information into our graph-based model framework, we extracted the item category attribute from the datasets, abstracting each unique category as an independent node and treating it as a relational link connecting users to items of the corresponding category. Table 1 summarizes the key statistical characteristics of the processed datasets. The behavioral distribution of the two datasets is consistent: browsing is the most frequent interaction, followed by adding to cart and purchasing. This realistic behavioral pattern ensures that our experiments are based on authentic user interaction data, providing a solid foundation for evaluating model performance.

Table 1. Statistics of the datasets.

Dataset	Users	Items	Views	Carts	Types	Buys
Taobao	15,449	11,953	873,954	195,476	2467	92,180
Tmall	41,731	11,953	1,813,498	223,510	1284	255,586

5.1.3. Evaluation Metrics

To rigorously assess the performance of our model, we adopted the leave-one-out evaluation strategy. Specifically, for each user, the most recent interaction was reserved as the test instance, while the remaining interactions were used as positive samples during training. In the evaluation phase, the model was required to rank candidate items based on their predicted relevance scores. For each positive sample derived from user interactions, we randomly selected a negative sample—an item the user had not interacted with—to compute the Bayesian Personalized Ranking (BPR) loss, which served as the optimization objective. To measure the effectiveness of the proposed approach in Top-N recommendation tasks, we employed two widely used metrics: Hit Ratio at 10 (HR@10) and Normalized Discounted Cumulative Gain at 10 (NDCG@10). The calculations of HR and NDCG are given by Equations (12) and (13) below.

$$\text{HR} = \frac{1}{S} \sum_{i=1}^S \text{hit}(i) \quad (12)$$

where S denotes the number of samples, which can be interpreted as the number of user demand items. The function $\text{hit}(i)$ indicates whether the i -th demand item is included in the item list recommended by the model: it takes a value of 1 if included, and 0 otherwise.

$$\text{NDCG} = \frac{1}{S} \sum_{i=1}^S \frac{1}{\log_2(p_i + 1)} \quad (13)$$

where S denotes the number of samples, which can be interpreted as the number of user demand items. $p(i)$ represents the position of the i -th demand item in the item list recommended by the model. If the i -th demand item is not in the recommendation list, the term $\frac{1}{\log_2(p_i + 1)}$ is set to 0.

5.1.4. Baseline Models

To demonstrate the performance of the proposed model, we selected several baseline methods from both single-behavior and multi-behavior recommendation systems for comparison. The details of these methods are as follows:

1. Single-behavior models:

NeuMF [55] is a traditional collaborative filtering model that integrates matrix factorization and multi-layer perceptron (MLP) to capture the nonlinear relationships between users and items; **LightGCN** [16] is a collaborative filtering model that aban-

dons the complex feature transformations and nonlinear activation functions inherent in graph convolutional networks (GCNs). It leverages multiple layers and higher-order graph convolutional operations to learn user and item representations, thereby capturing the latent association information between users and items. By simplifying the GCN model architecture, it achieves enhanced recommendation performance.

2. Multi-behavior models:

CRGCN [49] is a model that employs cascaded graph convolutional networks to model multi-behavior user data. It designs specific connection mechanisms between upstream and downstream behaviors to transmit information from upstream behaviors to downstream ones for modeling and incorporates multi-task learning to boost overall model performance. **MBCGCN** [37] utilizes a cascaded graph convolutional network structure to learn user and item embeddings by exploiting the behavioral dependencies within the user behavior chain, thus effectively improving recommendation accuracy. **NMTR** [56] combines neural networks with a multi-task learning framework to capture the cascaded relationships among diverse user behaviors (such as browsing, adding to cart, and purchasing) and optimize recommendation performance accordingly. **HPMR** [57] proposes a novel projection mechanism to explicitly model the correlations between upstream and downstream behaviors, which extracts valuable transferable information to facilitate the learning of downstream tasks while mitigating the negative transfer effect. **DMR** [42] combines disentangled representation and denoising modules to separate behavior-specific and shared features and suppresses noise interference in multi-behavior data via contrastive learning. **PKEF** [54] leverages the multi-behavior interaction information of users to tackle the problems of data distribution imbalance and negative transfer that are prevalent in multi-behavior recommendation scenarios.

5.1.5. Parameter Settings

In this experiment, we used the Adam optimizer for optimization, with the number of training epochs set to 100 and the node embedding dimension set to 64. The lr was adjusted within the range of 1×10^{-4} , 2×10^{-4} , 3×10^{-4} , 4×10^{-4} , 5×10^{-4} , 6×10^{-4} , and 7×10^{-4} . We initialized the model parameters, such as the initial node feature matrix, using the Xavier method. Additionally, we employed an early-stopping strategy. The number of layers in LightGCN was tuned within the range of 1, 2, and 3. In the fusion process of the attention mechanism, we tune the number of attention heads and the dimension of the attention hidden layer within the ranges of 1, 2, and 4 and 16, 32, 64, and 128, respectively, and conduct comparative experiments accordingly. We experimented with various behavior sequence relationships and ultimately determined the use of the view > cart > buy behavior sequence relationship chain in subsequent experiments. For the other baseline models, we utilized their open-source code, adjusted the model parameters, and compared their best performance publicly.

5.2. Main Results

This section compares our model with other baselines on Taobao and Tmall datasets using HR@10 and NDCG@10 as evaluation metrics, with results presented in Table 2 (our model's results are in bold, and the second-best results are underlined). The results show that models with multi-behavior modeling significantly outperform single-behavior models in recommendation accuracy, confirming the gain of integrating auxiliary behaviors (e.g., add-to-cart and view) for user preference learning. Single-behavior models relying solely on purchase behavior fail to achieve accurate recommendations due to data sparsity. Our model achieves the optimal performance on both datasets: HR@10 = 0.1540 and NDCG@10

= 0.0890 on Taobao; HR@10 = 0.1413 and NDCG@10 = 0.0815 on Tmall. Compared with the best-performing baseline, it achieves improvements of 10–11.1% (Taobao) and 9–10.6% (Tmall) in the two metrics, fully verifying the effectiveness of the proposed model.

Table 2. The overall performance comparison.

Dataset	Metric	Single-Behavior			Multi-Behavior					Ours
		NeuMF	LightGCN	CRGCN	MBCGCN	NMTR	HPMR	DMR	PKEF	
Taobao	HR@10	0.0241	0.0410	0.0886	0.1213	0.0319	0.1189	0.1254	<u>0.1368</u>	0.1540
	NDCG@10	0.0045	0.0261	0.0450	0.0703	0.0162	0.0613	0.0706	<u>0.0791</u>	0.0890
Tmall	HR@10	0.0218	0.0390	0.0846	0.0991	0.0540	0.0965	0.1056	<u>0.1286</u>	0.1413
	NDCG@10	0.0198	0.0206	0.0445	0.0557	0.0289	0.0519	0.0656	<u>0.0729</u>	0.0815

Among single-behavior models, NeuMF captures nonlinear relationships between behaviors via a multi-layer neural network, while LightGCN utilizes the user–item bipartite graph and GCN to obtain high-order neighbor information. It efficiently leverages high-order user–item correlations through neighborhood aggregation, thereby improving the accuracy of user interest capture. However, both fail to explore deep user preferences due to their reliance solely on purchase behavior.

Multi-behavior models have their own focuses but limitations: NMTR adopts multi-task joint training to model multi-behavior interactions, yet it ignores the difference in behavior importance, leading to unreasonable weight allocation and vulnerability to the negative impact of behaviors with scarce data or high noise. CRGCN, built on NMTR, incorporates sequential influences between multiple behaviors by passing upstream behavior embeddings to downstream for optimization. HPMR separates the shared and unique representations of upstream and downstream behaviors through a projection mechanism to reduce negative transfer. MBCGCN performs embedding learning along behavior chains to alleviate data sparsity and cold-start problems. DMR integrates disentangled representation and denoising modules for multi-behavior recommendation learning, decouples and separates behavior-specific and shared features, and effectively suppresses noise interference in multi-behavior data with contrastive learning. PKEF uses cascaded GCN to learn hierarchical correlations between behaviors and combines parallel learning to correct biases caused by imbalanced interaction distributions. Nevertheless, these models neither consider the multi-behavior patterns proposed in this paper nor the impact of interaction patterns between users and item types on preferences.

Our model adopts a cascaded structure with residual-connected LightGCN (similar to MBCGCN), which not only learns multi-behavior sequence information but also constructs diverse behavior patterns by dividing subgraphs to capture behavior interaction effects and the influence of item types. This design facilitates exploring deep user preferences and alleviates the sparsity of shopping behaviors. Results in Table 2 verify the effectiveness of the combined strategy of behavior chain learning and multi-behavior pattern learning.

5.3. Ablation Study

To verify the effectiveness of each module in our model, we conducted ablation experiments on the Taobao dataset (MP denotes multiple meta-path pattern learning and AF represents attention fusion in Table 3). The experiments were carried out in the following progressive steps: (1) Only considering buy behavior, we used ordinary GCN for single-behavior embedding learning. (2) Compared the performance of LightGCN and ordinary GCN. (3) Integrated auxiliary behaviors into LightGCN to construct a multi-behavior model. (4) Normalization was performed following the approach in Equation 8. (5) Combined LightGCN with our proposed multi-behavior patterns and fused subgraphs using mean

aggregation. (6) Replaced the fusion method with attention-based fusion. (7) Incorporated behavior sequence relationship learning to obtain the final model.

Table 3. Performances of ablation study.

Model	Taobao	
	HR@10	NDCG@10
(base) GCN (single-behavior)	0.0331	0.0212
LightGCN (single-behavior)	0.0410	0.0261
LightGCN + multi-behavior	0.1045	0.0637
LightGCN (with Equation 8) + multi-behavior	0.1267	0.0717
LightGCN + MP	0.1326	0.0789
LightGCN + MP + AF	0.1383	0.0811
(ours) LightGCN + MP + sequence + AF	0.1540	0.0890

The experimental results show that the single-behavior ordinary GCN performs poorly as it cannot capture potential interests using viewing and add-to-cart behaviors; replacing it with LightGCN leads to improvements of 23.9% and 18.8% in HR@10 and NDCG@10, respectively, proving that LightGCN is more suitable for capturing potential interests after removing complex nonlinear structures; integrating multi-behavior information results in significant increases of 67.6% and 63.6% in the two metrics, verifying the role of multi-behavior fusion in alleviating purchase behavior sparsity and mining potential interests. At the same time, after applying the method shown in Equation 8, the two evaluation metrics increased by 17.5% and 11.2%, respectively, compared to the scenario where this method was not used, which fully demonstrates the feasibility of the normalization method presented in Equation 8; fusing multi-behavior pattern subgraphs further improves the metrics by 4.4% and 9.1%, indicating that the defined multi-behavior patterns can obtain more accurate user-item information; replacing mean aggregation with attention fusion achieves a slight performance improvement, as it can accurately identify high-value information and assign weights; finally, incorporating behavior sequence learning brings additional improvements of 10.2% and 8.9% in the metrics, demonstrating that sequence relationship learning enables information transfer between upstream and downstream behaviors, and combining this with multi-behavior patterns enhances the accuracy of preference capture. In addition, upstream behaviors such as viewing may contain redundant information and cause negative impacts, which will be explored in future research.

5.4. Parameter Analysis

Table 4 summarizes the parameters and hyperparameters of our model. To ensure the reliability and reproducibility of experimental results, we systematically tuned the core hyperparameters (learning rate, training epochs, embedding dimension, number of attention heads, dimension of the attention hidden layer, and LightGCN layers). This tuning was performed on a held-out validation set, which was split from the original training set at a 9:1 ratio and kept fully independent of the test set to prevent overfitting. The parameter experimental results on the Taobao and Tmall datasets are illustrated in Figures 4 and 5, respectively, with detailed experimental content and analyses presented in the following four subsections.

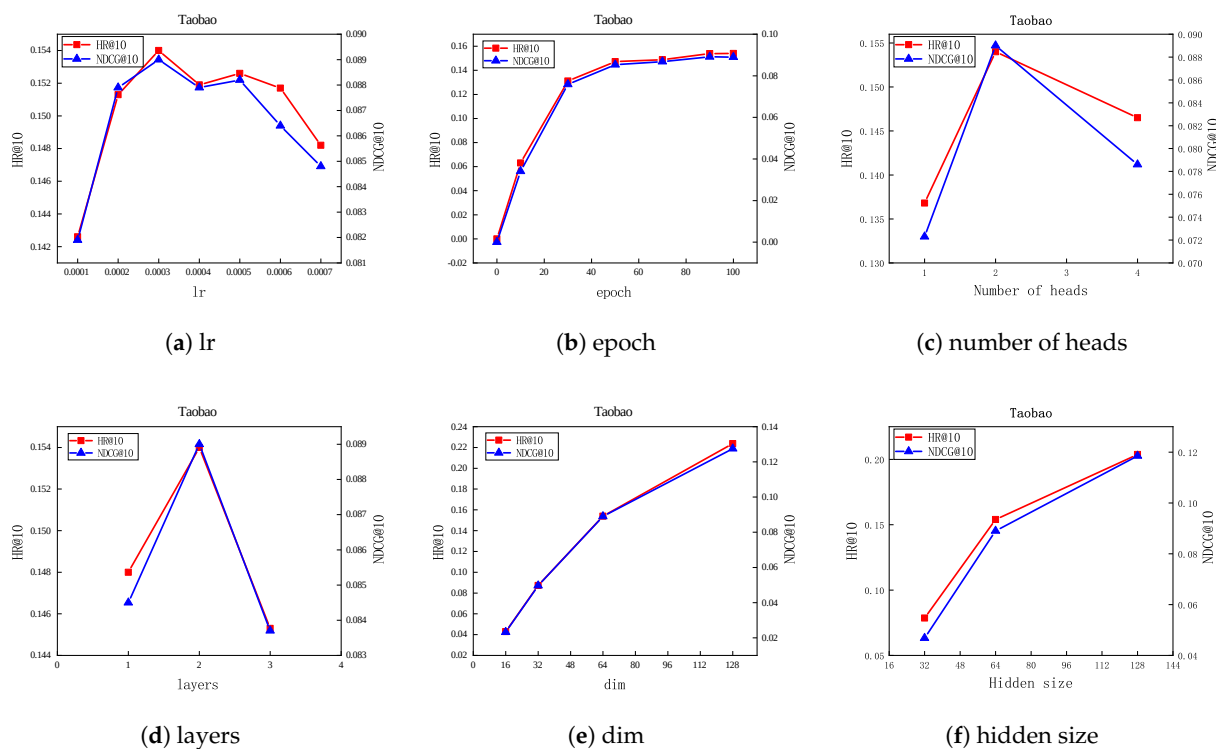


Figure 4. The impact of four parameters on the Taobao dataset.

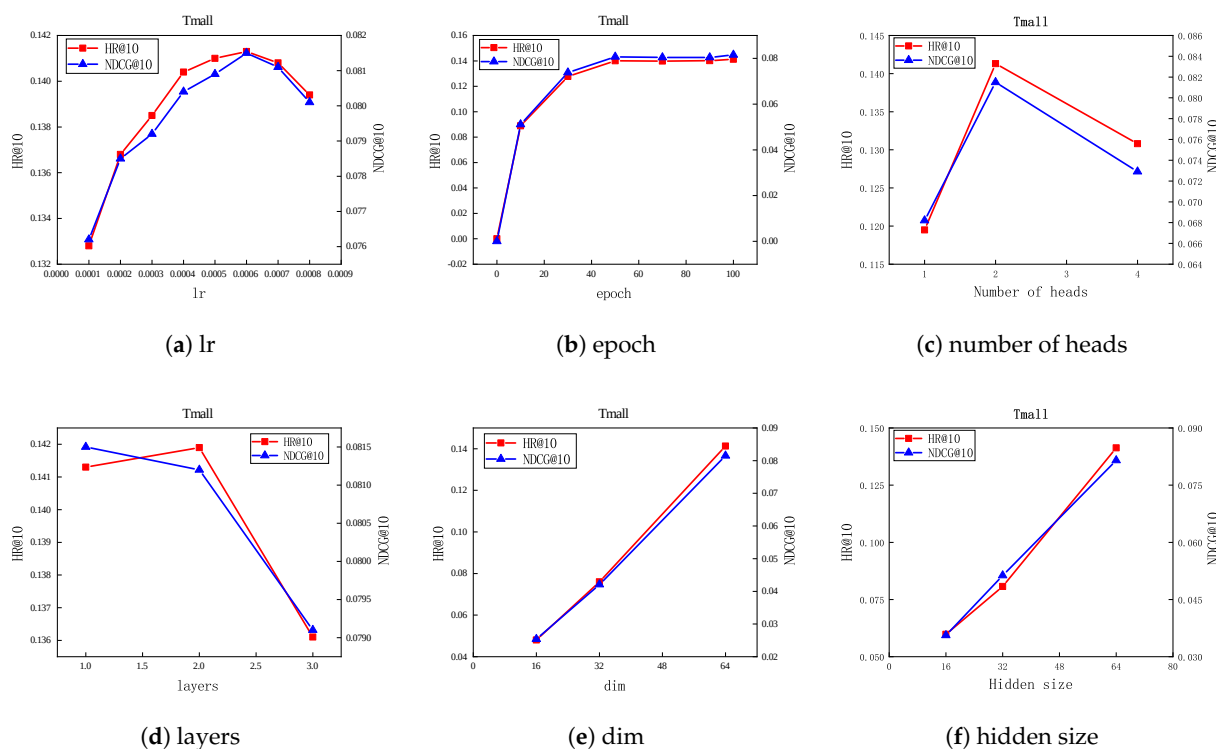


Figure 5. The impact of four parameters on the Tmall dataset.

Table 4. Definitions of key parameters and hyperparameters.

Parameter	Description
k	Behavior
σ_k	Learnable weight coefficient for behavior k
$\mathbf{a}^\top$	Learnable attention vector
λ	Weight coefficient controlling L_2 regularization
l	Layers of LightGCN
lr	Learning rate
epoch	Training epochs
dim	Embedding dimension

5.4.1. Learning Rate

We searched for the optimal learning rate size within the range of 0.0001, 0.0002, 0.0003, 0.0004, 0.0005, 0.0006, and 0.0007. Different datasets should use different learning rates. As can be seen from Figures 4 and 5, the optimal results were achieved on the Taobao and Tmall datasets when the learning rates were 0.0003 and 0.0006, respectively. On the Taobao and Tmall datasets, when the lr is less than 0.0003 and 0.0006, respectively, the model might fail to converge to the optimal solution, thereby affecting model performance. When the lr exceeds 0.0003 and 0.0006, respectively, the model might overfit, impacting its generalization ability. The experimental results demonstrated that the optimal lr for the Taobao dataset was 0.0003, and for the Tmall dataset, it was 0.0006.

5.4.2. Training Epoch

We searched for the optimal number of training epochs within the range of 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100. As shown in Figures 4 and 5, the model's HR@10 and NDCG@10 increased rapidly during the initial training phase. After 40–50 epochs of training, the growth gradually slowed down. In the initial stage, the model can quickly learn the main patterns and features in the data, rapidly improving the accuracy of predictions. As the number of training epochs increases, the model parameters gradually approach the optimal solution, the gradients decrease, the parameter update magnitudes become smaller, and the performance improvement also stabilizes. Therefore, we chose 100 epochs as the training round for this model.

5.4.3. LightGCN Layers

We searched for the optimal number of LightGCN layers within the range of 1, 2, and 3. As can be seen from Figures 4 and 5, the optimal results were achieved on the Taobao and Tmall datasets when the number of LightGCN layers was two. When the number of LightGCN layers was one, the results were close to those with two layers, achieving the second-best performance. However, when the number of layers was three, the performance was the worst. Combining the theoretical content, we know that one-layer GCN can only aggregate information from direct neighbor nodes. Although it can capture local neighborhood information, the scope of information propagation is limited, making it difficult to capture broader graph-structured information. Two-layer GCN can aggregate information from two-hop neighbor nodes, capturing more extensive graph-structured information. three-layer GCN may lead to over-smoothing and overfitting, resulting in poor experimental outcomes and an inability to generalize to new data. Therefore, to capture higher-level information and better adapt to multiple datasets, we chose a two-layer GCN to learn the graph-structured information and obtain user and item node embeddings.

5.4.4. Embedding Dimension

We searched for the optimal embedding dimension within the range of 16, 32, 48, 64, and 128. The results are shown in Figures 4 and 5. Theoretically, as the dimension increases, the node information becomes richer, allowing the model to better capture user interaction information, identify user preferences, and achieve accurate recommendations. However, increasing the embedding dimension also leads to higher computational costs, increased device pressure, and more redundant information. The experimental results indicate that model performance continues to improve with an increasing embedding dimension, showing that the model can obtain information in higher dimensions. Considering the experimental cost and model performance, we chose 64 as the node embedding dimension.

5.4.5. Number of Heads

We searched for the optimal number of attention heads within the range of 1, 2, and 4, and the experimental results are shown in Figures 4 and 5. Theoretically, an increase in the number of attention heads enables the model to better capture the latent information under different edge types. However, a larger number of attention heads introduces more redundancy and simultaneously increases the model training time. Based on the experimental results of the number of attention heads on both the Taobao and Tmall datasets, we finally set the number of attention heads to two as the optimal configuration for our model.

5.4.6. Hidden Size

We searched for the optimal dimension of the attention mechanism's hidden layer within the range of 16, 32, 64, and 128, and the experimental results are shown in Figures 4 and 5. For the Tmall dataset, the training time at the 128-dimension setting was excessively long, so the experiments were only conducted within the range of 16, 32, and 64 for this dataset. Theoretically, a higher dimension for user and item embeddings enables the attention mechanism to capture more latent information and better distinguish whether users are interested in items. However, an increase in dimension leads to an exponential rise in training time. Based on the analysis of the experimental results, we finally selected 64 as the dimension of the attention mechanism's hidden layer.

5.5. Multi-Behavior Experiment

To verify the effectiveness of the multi-behavior patterns incorporating user interactions designed in this paper (Table 5), we conducted three progressive experiments on the Taobao dataset. In the first stage, only buy behavior was used for learning; no multiple meta-path patterns were designed due to the absence of multiple behavior types, resulting in poor performance with HR@10 = 0.0335 and NDCG@10 = 0.0197. The core reason is that single-behavior modeling fails to capture deep user preferences.

Table 5. Performances of multi-behavior combinations .

Behavior	Taobao	
	HR@10	NDCG@10
only buy	0.0335	0.0197
cart + buy	0.1040	0.0657
view + cart + buy	0.1540	0.0890

In the second stage, cart behavior was added, and multiple meta-path patterns (e.g., $U \xrightarrow{\text{cart, buy}} I \xrightarrow{\text{cart}} U$) were set up to learn user interactions under specific patterns through subgraphs. Compared with the first stage, HR@10 and NDCG@10 significantly increased by

67.8% and 70%, respectively. The introduction of cart behavior expanded the scope of user interest capture (including potential purchase items), while multiple meta-path patterns helped the model focus on high-value information and reduce irrelevant interference.

In the third stage, view behavior was added on the basis of the second stage, with additional similar multiple meta-path patterns. HR@10 and NDCG@10 further improved by 32.5% and 26.2% compared with the second stage. The incorporation of view behavior enabled the model to capture more nuanced user interests (including items only viewed without conversion).

Experiments demonstrate that gradually integrating user behaviors can comprehensively explain preferences and improve recommendation accuracy; multiple meta-path patterns can provide rich information propagation paths, accurately obtain effective information, and reduce noise impact, thereby optimizing model performance.

5.6. Sensitivity Analysis

We conduct a sensitivity analysis on the parameter σ_k mentioned in Equation (3). The initial parameter σ_k is determined via random initialization, and a set of reasonable parameter values is derived through model training. As shown in Table 6, we assign different weight coefficients to the four types of edges corresponding to view, add-to-cart, buy, and product item category, which are {0.1, 0.2, 0.5, 0.2}, {0.5, 0.2, 0.1, 0.2}, {0.1, 0.5, 0.1, 0.3}, and the randomly initialized weight coefficients, respectively. From the results in the table, it can be concluded that the experimental outcomes are barely affected by either the method based on Equation (3) or the random initialization of weights, demonstrating the robustness of our approach.

Table 6. The analysis of σ_k .

Initialization Strategy	HR@10	NDCG@10
0.1, 0.2, 0.5, 0.2	0.1536	0.0882
0.5, 0.2, 0.1, 0.2	0.1526	0.0856
0.1, 0.5, 0.1, 0.3	0.1531	0.0879
random	0.1540	0.0890

5.7. Behavior Sequence Relationship Experiments

To evaluate the impact of different behavior sequence relationships in multi-behavior scenarios, we conducted experiments on the Taobao dataset (results shown in Figure 6). The sequence learning adopted the approach of “using upstream behavior embeddings as downstream inputs + residual connections to retain information”, and five sequences with the same types and number of behaviors were compared: VBC (view, buy, cart), CBV (cart, buy, view), BVC (buy, view, cart), BCV (buy, cart, view), and VCB (view, cart, buy).

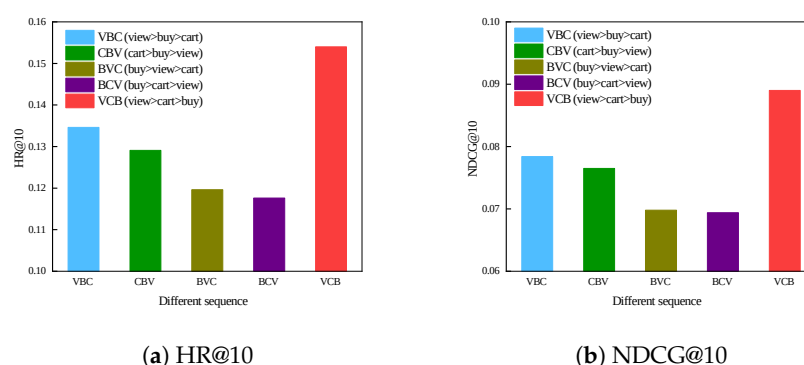


Figure 6. Results under different behavior sequences.

The results indicate that BVC and BCV perform the worst (HR@10: 0.1196, 0.1176; NDCG@10: 0.0698, 0.0694) because they violate the natural user shopping logic (rare to view/add to cart after purchase), contain substantial noise, and make it difficult for the model to learn effective information. CBV and VBC show moderate performance, with improvements of 8.9–9.3% and 14.5–11.5% over BCV, respectively; although they have certain behavioral logical consistency, they still contain unfilterable noise. VCB (adopted in our model) achieves the optimal performance, with HR@10 = 0.1540 and NDCG@10 = 0.0890, representing increases of 23.6% and 22.0% compared to BCV.

VCB conforms to the natural user shopping logic of “view–cart–buy”, which can truly reflect behavioral paths and preferences, providing rich context for the model to accurately capture user intentions. Combined with the model’s learning process, it can alleviate data sparsity and reduce noise impact. The experimental results further verify the effectiveness of integrating multiple meta-path patterns and behavior sequences in our model.

6. Conclusions

In this paper, we present a multi-behavior and sequence-aware recommendation method. It models sequential correlations among diverse user behaviors while exploiting structural dependencies induced by multiple meta-path patterns. By incorporating Light-GCN with residual connections, the proposed model effectively learns user neighborhood representations, captures both user–user and user–item interaction dependencies, and thereby achieves more accurate recommendation performance.

Comprehensive experiments on two real-world datasets, including performance evaluation, ablation studies, parameter sensitivity analysis, and investigations of multi-behavior and sequential effects, confirmed the effectiveness of the proposed approach. The findings demonstrate that incorporating multiple meta-path patterns allows the model to capture more discriminative information, improving recommendation accuracy while reducing redundancy.

For future work, we aim to further refine the design of meta-path patterns by constructing finer-grained subgraphs that account for temporal dynamics in user interactions, enabling more precise modeling of evolving user preferences. Additionally, we plan to validate the real-world applicability of the proposed method through large-scale online A/B testing.

Author Contributions: Conceptualization, D.Y. and T.W.; methodology, D.Y. and T.W.; data curation, T.W.; formal analysis, T.W.; investigation, T.W.; writing—original draft, D.Y. and T.W.; writing—review & editing, D.Y. and T.W.; visualization, D.Y. and T.W.; supervision, D.Y.; project administration, D.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Outstanding Young Talent Program of Beijing Municipal Universities of No. BPHR202203081.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: These data were derived from the following resources available in the public domain: Taobao was derived from the research entitled [54]. The associated data resources are available in the GitHub repository <https://github.com/MC-CV/PKEF> (accessed on 1 February 2026). Tmall was derived from the research entitled [37]. The associated data resources are available in the GitHub repository <https://github.com/SS-00-SS/MBCGCN> (accessed on 1 February 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Kim, K.; Kim, S.; Lee, G.; Jung, J.; Shin, K. Multi-Behavior Recommender Systems: A Survey. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 435–452.
- Li, F.; Liu, S.; Wang, H. Collaborative filtering or content-based recommendation? The effects of digital platform recommendation type on consumer's intention. *J. Retail. Consum. Serv.* **2026**, *90*, 104694.
- Wang, X.; He, X.; Cao, Y.; Liu, M.; Chua, T.S. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, 4–8 August 2019; pp. 950–958.
- Fu, M.; Zhang, H.; Du, Y.; Luo, R. Adaptive Attention-based State Representation in reinforcement learning based recommendation systems. *J. Eng. Res.* **2026**, 10.1016/j.jer.2026.01.012.
- Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; Xie, X. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Virtual, Canada, 11–15 July 2021; pp. 726–735.
- Yu, J.; Yin, H.; Xia, X.; Chen, T.; Cui, L.; Nguyen, Q.V.H. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Madrid, Spain, 11–15 July 2022; pp. 1294–1303.
- Gu, A.; Gulcehre, C.; Paine, T.; Hoffman, M.; Pascanu, R. Improving the gating mechanism of recurrent neural networks. In *Proceedings of the International Conference on Machine Learning*, PMLR, Virtual, 13–18 July 2020; pp. 3800–3809.
- Qiu, Z.; Wang, Z.; Zheng, B.; Huang, Z.; Wen, K.; Yang, S.; Men, R.; Yu, L.; Huang, F.; Huang, S.; et al. Gated Attention for Large Language Models: Non-linearity, Sparsity, and Attention-Sink-Free. *arXiv* **2025**, arXiv:2505.06708.
- Arevalo, J.; Solorio, T.; Montes-y Gómez, M.; González, F.A. Gated multimodal units for information fusion. *arXiv* **2017**, arXiv:1702.01992.
- Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24.
- Han, Z.; Zhou, T.; Chen, G.; Chen, J.; Fu, C. A Robust Rating Prediction Model for Recommendation Systems Based on Fake User Detection and Multi-Layer Feature Fusion. *Big Data Min. Anal.* **2025**, *8*, 292–309.
- Zhang, M.; Zhang, X.; Pedrycz, W.; Wang, S.; Wu, G. Learning Fine-Grained User Preference for Personalized Recommendation. *Tsinghua Sci. Technol.* **2025**, *30*, 2544–2556.
- Mao, K.; Zhu, J.; Xiao, X.; Lu, B.; Wang, Z.; He, X. UltraGCN: ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, Queensland, Australia, 1–5 November 2021; pp. 1253–1262.
- Wang, X.; He, X.; Wang, M.; Feng, F.; Chua, T.S. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Paris, France, 21–25 July 2019; pp. 165–174.
- Guo, L.; Hua, L.; Jia, R.; Zhao, B.; Wang, X.; Cui, B. Buying or browsing?: Predicting real-time purchasing intent using attention-based deep network with multiple behavior. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, 4–8 August 2019; pp. 1984–1992.
- He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Virtual, China, 25–30 July 2020; pp. 639–648.
- Liu, F.; Cheng, Z.; Zhu, L.; Gao, Z.; Nie, L. Interest-aware message-passing GCN for recommendation. *Proc. Web Conf.* **2021**, *2021*, 1296–1305.
- Xia, L.; Huang, C.; Xu, Y.; Dai, P.; Zhang, B.; Bo, L. 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*, Virtual, China, 25–30 July 2020; pp. 2397–2406.
- Xia, L.; Huang, C.; Xu, Y.; Dai, P.; Zhang, X.; Yang, H.; Pei, J.; Bo, L. Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 4486–4493.
- Wang, J.; Cao, Y.; Zhang, F.; Kou, F.; Wei, K.; Zhang, J.; Chen, J. IDBR: Interaction-Aware Dual-Granularity Learning for Bundle Recommendation. *Big Data Min. Anal.* **2025**, *8*, 751–766.
- Sang, L.; Yuan, H.; Huang, Y.; Zhang, Y. Graph structure learning for robust recommendation. *Tsinghua Sci. Technol.* **2024**, *30*, 1617–1635.
- Hu, L.; Song, G.; Xie, Z.; Zhao, K. Personalized recommendation algorithm based on preference features. *Tsinghua Sci. Technol.* **2014**, *19*, 293–299.
- Chen, C.; Ma, W.; Zhang, M.; Wang, Z.; He, X.; Wang, C.; Liu, Y.; Ma, S. Graph heterogeneous multi-relational recommendation. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 3958–3966.

24. Jin, B.; Gao, C.; He, X.; Jin, D.; Li, Y. Multi-behavior recommendation with graph convolutional networks. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, China, 25–30 July 2020; pp. 659–668.
25. Meng, C.; Zhao, Z.; Guo, W.; Zhang, Y.; Wu, H.; Gao, C.; Li, D.; Li, X.; Tang, R. Coarse-to-fine knowledge-enhanced multi-interest learning framework for multi-behavior recommendation. *ACM Trans. Inf. Syst.* **2023**, *42*, 1–27.
26. Xia, L.; Xu, Y.; Huang, C.; Dai, P.; Bo, L. Graph meta network for multi-behavior recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 11–15 July 2021; pp. 757–766.
27. Bellini, P.; Palesi, L.A.I.; Nesi, P.; Pantaleo, G. Multi clustering recommendation system for fashion retail. *Multimed. Tools Appl.* **2023**, *82*, 9989–10016.
28. Li, T.; Wang, J.; Ye, F.; Feng, X.; Zhang, Y. A Collaborative Filtering Recommendation Algorithm Based on User Clustering. *Syst. Eng. Electron.* **2007**, *29*, 1178–1182.
29. Deng, A.; Zuo, Z.; Zhu, Y. A Collaborative Filtering Recommendation Algorithm Based on Item Clustering. *J. Chin. Comput. Syst.* **2004**, *25*, 1665–1670.
30. Cheng, G.; Liu, X.; Chen, D. A Personalized Recommendation Algorithm Based on Hybrid Clustering. *J. Wuhan Univ. Technol. Inf. Manag. Eng. Ed.* **2011**, *33*, 379–381.
31. Lan, W.; Zhou, G.; Chen, Q.; Wang, W.; Pan, S.; Pan, Y.; Zhang, S. Contrastive clustering learning for multi-behavior recommendation. *ACM Trans. Inf. Syst.* **2024**, *43*, 1–23.
32. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* **2012**, arXiv:1205.2618.
33. Gao, C.; He, X.; Gan, D.; Chen, X.; Feng, F.; Li, Y.; Chua, T.S.; Jin, D. Neural multi-task recommendation from multi-behavior data. In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE); IEEE, Piscataway, NJ, USA: 2019; pp. 1554–1557.
34. Lian, J.; Zhou, X.; Zhang, F.; Chen, Z.; Xie, X.; Sun, G. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining London, UK, 19–23 August 2018; pp. 1754–1763.
35. Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; Gai, K. Deep interest network for click-through rate prediction. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1059–1068.
36. Addula, S.R.; Tyagi, A.K. Future of computer vision and industrial robotics in smart manufacturing. *Artif.-Intell.-Enabled Digit. Twin Smart Manuf.* **2024**, 505–539, 10.1002/9781394303601.ch22.
37. Cheng, Z.; Han, S.; Liu, F.; Zhu, L.; Gao, Z.; Peng, Y. Multi-behavior recommendation with cascading graph convolution networks. In Proceedings of the ACM Web Conference 2023, Austin, TX, USA, 30 April–4 May 2023; pp. 1181–1189.
38. Singh, A.P.; Gordon, G.J. Relational learning via collective matrix factorization. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 650–658.
39. Krohn-Grimberghe, A.; Drumond, L.; Freudenthaler, C.; Schmidt-Thieme, L. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, Seattle, WA, USA, 8–12 February 2012; pp. 173–182.
40. Qiu, H.; Liu, Y.; Guo, G.; Sun, Z.; Zhang, J.; Nguyen, H.T. BPRH: Bayesian personalized ranking for heterogeneous implicit feedback. *Inf. Sci.* **2018**, *453*, 80–98.
41. Chen, Y.; Cao, J.; Wang, Y.; Wu, J.; Chen, H.; Xu, G. Causal variational inference for deconfounded multi-behavior recommendation. *ACM Trans. Inf. Syst.* **2025**, *43*, 1–26.
42. Zhang, Y.; Chen, W.; Cai, F.; Shi, Z.; Qi, F. DMR: disentangled and denoised learning for multi-behavior recommendation. *Complex Intell. Syst.* **2025**, *11*, 153.
43. Huang, S.; Li, Q.; Wang, X.; Yu, D.; Xu, G.; Li, Q. Counterfactual Debasing for Multi-behavior Recommendations. In Proceedings of the International Conference on Database Systems for Advanced Applications; Springer: Berlin/Heidelberg, Germany, 2024; pp. 164–179.
44. Zhang, W.; Mao, J.; Cao, Y.; Xu, C. Multiplex graph neural networks for multi-behavior recommendation. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, 19–23 October 2020; pp. 2313–2316.
45. Chen, H.; Li, Z.; Bei, Y.; Xu, K.; Zhang, Y.; Huang, F.; Yang, Y.; Gong, H.; Karray, F. Behavior Merging Graph Convolution Network for Multi-Behavior Recommendation. *IEEE Trans. Knowl. Data Eng.* **2025**, *37*, 6987–7000.
46. Yang, H.; Chen, H.; Li, L.; Yu, P.S.; Xu, G. Hyper meta-path contrastive learning for multi-behavior recommendation. In Proceedings of the 2021 IEEE International Conference on Data Mining (ICDM); IEEE, Piscataway, NJ, USA: 2021; pp. 787–796.
47. Choi, J.; Kwon, J.; Kim, Y.; Kim, Y. Hypergraph temporal multi-behavior recommendation. *Eng. Appl. Artif. Intell.* **2025**, *145*, 110112.

48. Liu, Y.; Rao, Q.; Pan, W.; Ming, Z. Variational collective graph autoencoder for multi-behavior recommendation. In *Proceedings of the 2023 IEEE International Conference on Data Mining (ICDM)*; IEEE, Piscataway, NJ, USA: 2023; pp. 438–447.
49. Yan, M.; Cheng, Z.; Gao, C.; Sun, J.; Liu, F.; Sun, F.; Li, H. Cascading residual graph convolutional network for multi-behavior recommendation. *ACM Trans. Inf. Syst.* **2023**, *42*, 1–26.
50. Yu, X.; Ren, X.; Sun, Y.; Gu, Q.; Sturt, B.; Khandelwal, U.; Norick, B.; Han, J. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, New York, NY, USA, 24–28 February 2014; pp. 283–292.
51. Xie, F.; Zheng, A.; Chen, L.; Zheng, Z. Attentive meta-graph embedding for item recommendation in heterogeneous information networks. *Knowl.-Based Syst.* **2021**, *211*, 106524.
52. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 10.48550/arXiv.1706.03762. .
53. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth International Conference on Artificial Intelligence and Statistics*, 2010; pp. 249–256.
54. Meng, C.; Zhai, C.; Yang, Y.; Zhang, H.; Li, X. Parallel knowledge enhancement based framework for multi-behavior recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, Birmingham, UK, 21–25 October 2023; pp. 1797–1806.
55. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.S. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, Perth, Australia, 3–7 April 2017; pp. 173–182.
56. Gao, C.; He, X.; Gan, D.; Chen, X.; Feng, F.; Li, Y.; Chua, T.S.; Yao, L.; Song, Y.; Jin, D. Learning to recommend with multiple cascading behaviors. *IEEE Trans. Knowl. Data Eng.* **2019**, *33*, 2588–2601.
57. Meng, C.; Zhang, H.; Guo, W.; Guo, H.; Liu, H.; Zhang, Y.; Zheng, H.; Tang, R.; Li, X.; Zhang, R. Hierarchical projection enhanced multi-behavior recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Long Beach, CA, USA, 6–10 August 2023; pp. 4649–4660.

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