Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.076137

ARTICLE

Diverse Behavior Path Graphs for Multi-Behavior Recommendation

Qian Hu, Lei Tan*, Qingjun Yuan, Zong Zuo and Yan Li

Henan Key Laboratory of Cyberspace Situation Awareness, Zhengzhou, China

*Corresponding Author: Lei Tan. Email: tanlei08@sina.com

Received: 14 November 2025; Accepted: 15 January 2026; Published: 09 April 2026

ABSTRACT: Multi-behavior recommendation methods leverage various types of user interaction behaviors to make personalized recommendations. Behavior paths formed by diverse user interactions reveal distinctive patterns between users and items. Modeling these behavioral paths captures multidimensional behavioral features, which enables accurate learning of user preferences and improves recommendation accuracy. However, existing methods share two critical limitations: (1) Lack of modeling for the diversity of behavior paths; (2) Ignoring the impact of item attribute information on user behavior paths. To address these issues, we propose a Directed Behavior path graph-based Multi-behavior Recommendation method (DBMR). Specifically, we first construct a directed user-item behavior path graph based on diverse behavior chains. For each behavior, we then build a user-item interaction graph and use a LightGCN model with residual design to learn user and item embeddings. Next, we introduce a graph attention message aggregator that integrates features from previous behaviors into the learning of the next behavior, weighted by the transition strength between behaviors. Finally, we compute the recommendation score from the user preference and item representations under the target behavior. We adopt a joint optimization framework with a multi-task learning strategy, which accounts for each auxiliary behavior's contribution to target behavior prediction. Additionally, an auxiliary loss measures the difference between item embeddings from behavior paths and those from an attribute-feature encoder, thereby capturing multidimensional item features and refining recommendation results. Experiments on two real-world datasets demonstrate the effectiveness of our method in utilizing multi-behavior data.

KEYWORDS: Multi-behavior recommendation; diverse behavior path graphs; joint optimization; graph attention networks

1 Introduction

In recent years, recommendation systems (RS) have played a crucial role in alleviating information overload across personalized services such as e-commerce, social media, and video platforms. Their core objective is to identify user preferences and predict potential interactions from a vast pool of candidate items. Collaborative Filtering (CF)-based methods [1–3], as the most widely adopted approach, have achieved significant success. However, traditional CF models are typically designed for a single interaction type (e.g., purchases). Consequently, if a new user lacks historical purchase records, a model relying solely on purchase data may fail to provide effective recommendations, leading to the cold-start problem. In reality, user-item historical data often exhibits complex and diverse characteristics [4,5]. To fully leverage this richness, multi-behavior recommendation [6] has garnered widespread attention. It aims to predict a target behavior (e.g., purchase) by utilizing other observed behaviors as auxiliary signals [7]. Multi-behavior recommendation enriches user preference modeling by leveraging various interaction types (such as *page view*, *add to cart*, and *purchase*). These behaviors often follow a natural, predefined sequence: *page view* → *add to cart* → *purchase*

(as shown by the solid arrows in Fig. 1). Lower-level behaviors (e.g., *view*) provide valuable auxiliary signals for predicting higher-level ones (e.g., *purchase*), thereby mitigating data sparsity and alleviating the cold-start issue. Furthermore, the diverse behavior paths encapsulate latent user-specific interaction patterns. Modeling these paths facilitates a more comprehensive understanding of user preferences, ultimately leading to more accurate recommendations.

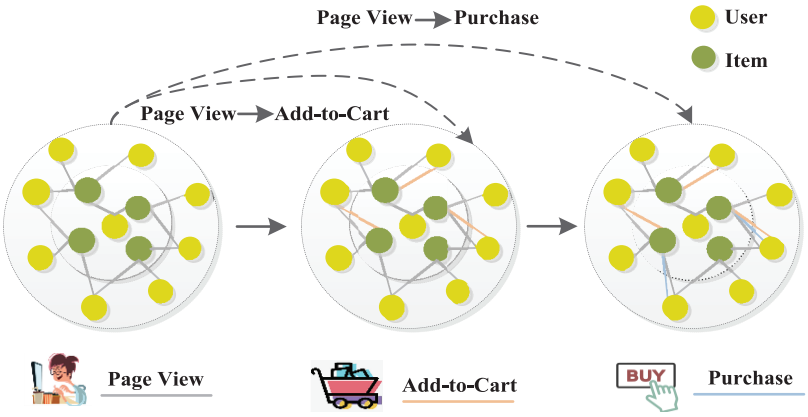

Figure 1: User-item interaction diagram with multiple behaviors.

Current research in multi-behavior recommendation primarily follows two paradigms for modeling behavioral diversity and dependencies: parallel and cascading approaches. Parallel methods integrate multiple single-behavior interaction graphs into a unified heterogeneous graph. They then extract and aggregate semantic and contextual information from different behaviors concurrently to capture behavioral dependencies. In contrast, cascading methods model the partial order within behavioral chains by sequentially connecting different behaviors, often using modules such as residual connections. However, most existing cascading methods rely on simplified assumptions about behavioral relationships. Consequently, they often fail to capture the complex path patterns present in real-world scenarios, which is primarily reflected in:

1. Lack of modeling behavioral path diversity. Most studies assume user behaviors follow a fixed, monotonous chain (e.g., *page view* → *add to cart* → *purchase*) [8]. This assumption overlooks the diversity and non-linearity of real-world user behaviors. As shown in Fig. 1, the graph depicts user–item (inner ring) and item–user (outer ring) interactions. The dashed lines indicate the existence of multiple possible behavioral paths, which together form a rich multi-behavior interaction structure. In reality, paths leading to a purchase can vary significantly: some users may skip intermediate steps (e.g., adding to cart) and proceed directly to purchase, while other behavioral paths may not result in a purchase at all.

2. Ignoring the impact of item attributes. Existing models typically aggregate semantics from different user behaviors to learn user preferences, without considering the influence of item attributes on behavior paths. For example, when purchasing everyday necessities such as toothpaste or bottled water, users tend to purchase directly and skip intermediate steps. However, when purchasing luxury goods such as diamonds, people are more likely to add the item to their cart first before purchasing, in order to consider their choice carefully. This shows that inherent item attributes, such as attractiveness and price, influence user behavior paths. These attributes are indirectly reflected through how items are engaged with across different behaviors.

To address the above limitations, we propose a Directed Behavior path graph-based Multi-behavior Recommendation method (DBMR). Specifically, we construct directed behavior path graphs for both users and items based on behavior chains. We first adopt a LightGCN model with a residual design on the user-item interaction graph for each behavior to integrate the information from adjacent nodes into the

embeddings, thereby learning the feature embeddings for both users and items for each behavior. Afterward, we propose a graph attention aggregator to weight-aggregate the features learned from previous behaviors into the feature learning for the next behavior. Subsequently, the recommendation score is calculated based on the user preference and item characteristic representations under the target behavior, thereby predicting the likelihood of a user purchasing the item. Finally, during model optimization, we introduce personalized multi-task learning (MTL), where the co-occurrence scores of user-item interactions across different behaviors are computed to quantify the contribution of each auxiliary behavior to the prediction of the target behavior. Simultaneously, an Attribute Feature Encoder is employed to learn item attribute embeddings, which are then used to reconstruct the item behavior embeddings learned from the behavior paths. In summary, the contributions of this paper can be outlined as follows:

- We propose a multi-behavior recommendation method based on a directed behavior path graph. It effectively aggregates user and item feature embeddings across different behaviors, based on varying transition intensities. By updating user preferences and item characteristics through behavioral chains, the method directly incorporates behavioral dependencies into the embedding learning process.
- We comprehensively consider the impact of diverse behavior paths and item attributes on user behavior sequences. A directed behavior path graph is constructed to support the modeling of skip behaviors and non-sequential paths. Meanwhile, semantic alignment constraints based on item attributes are established to optimize the learning of item representations.
- To validate the effectiveness of DBMR, we conduct extensive experiments on two real-world datasets. The results demonstrate that our model outperforms state-of-the-art baselines in recommendation performance.

2 Related Work

2.1 Single-Behavior Recommendation

It recommends items based on a single type of user behavior, such as ratings [9], clicks [10], trust [11], likes [12]. The early models utilized matrix factorization (MF) [9], graph-based models [13], and ranking based on random walks [14]. MF-BPR [10] was proposed to model interactive behaviors by combining the Bayesian Personalized Ranking (BPR) loss with matrix factorization. NCF [1] integrates traditional matrix factorization techniques with a multilayer perceptron to learn nonlinear features of user-item interactions. By combining linear interactions in a low-dimensional space and nonlinear interactions in a high-dimensional space, it improves the accuracy of predicting user preferences for items. However, the recommendation capability of single-behavior is constrained by the issue of data sparsity, leading to insufficient learning capacity [15,16].

2.2 Multi-Behavior Recommendation

To address the limitations of single-behavior recommendation, multi-behavior recommendation systems leverage multiple auxiliary behaviors (e.g., *page view*, *add to cart*) to predict the target behavior (e.g., *purchase*). Early multi-behavior methods were primarily extended from traditional recommendation techniques. A straightforward approach is to adapt traditional matrix factorization (MF) from operating on a single interaction matrix to multiple matrices [17]. Current methods can be broadly categorized by their behavior modeling strategy into two types: (1) parallel methods and (2) cascading methods.

1. Parallel Methods. This approach learns embeddings from each behavior graph independently to address the data imbalance across different interaction types, thereby reducing bias from dominant behaviors. MB-HGCN [18] utilizes global embeddings from a unified heterogeneous graph to initialize

graph convolutional networks (GCNs) for individual behaviors, allowing its multi-task learning (MTL) module to effectively leverage the refined embeddings. HPMR [19] extracts behavior embeddings in parallel and decomposes them into unique and shared components via projection, which helps mitigate negative transfer in MTL [20]. Meng et al. [21] argue that behavioral imbalance exacerbates negative transfer in cascade schemes and thus propose a projection-based distillation technique to integrate parallel and cascade strategies.

2. Cascade Methods. These methods model the sequential dependencies among user behaviors. ChainRec [22] extends matrix factorization (MF) by incorporating cascade dependencies and optimizing the scoring function. NMTR [23] utilizes MF to adapt multi-task learning (MTL) for predicting each behavior within a cascade sequence. CRGCN [24] introduces a cascaded Graph Convolutional Network (GCN) with residual connections to propagate behavior embeddings sequentially within an MTL framework. MB-CGCN [25] encodes nodes using LightGCN for each behavior and transmits features along the cascade chain via feature transformation. However, due to the imbalanced distribution of interactions across behaviors, a simple cascade network may allow high-frequency behaviors to dominate the learning process, thereby interfering with downstream behavior prediction. DA-GCN [26] proposes a Directed Acyclic Graph Convolutional Network for multi-behavior recommendation. It extracts collaborative signals by modeling partial-order relationships in monotonic behavior chains and employs an attention aggregator to gather information from potential preceding behaviors for subsequent feature learning. BCIPM [27] learns behavior-contextual item representations through a dedicated network, focusing only on target-behavior-related preferences to reduce auxiliary behavior noise. PPN-ARE [28] addresses high-order graph learning in cascades by using residual connections to simulate user preference evolution. It constructs positive and negative feedback chains from multi-behavior sequences to learn multi-level user interests. CMSR [29] is a causal multi-behavior sequential recommendation framework that uses a cascade network to capture directional dependencies within behavior chains. CGCL [30] introduces a cascaded graph contrastive learning framework that learns differentiated user representations for each behavior type by iteratively propagating preferences through the cascade structure. In this paper, we integrate parallel and cascade approaches by generalizing monotonic behavior chains into directed behavior path graphs. This allows behavior embeddings to be learned from multiple interaction graphs while progressively updating user preferences and item characteristics along behavioral paths, thereby more effectively capturing behavioral dependencies.

3 Methodology

3.1 Problem Setup and Formulation

User-item interaction. Let $\mathcal{U} = \{u_1, u_2, \cdots, u_{|\mathcal{U}|}\}, \mathcal{I} = \{i_1, i_2, \cdots, i_{|\mathcal{I}|}\}$ denote the set of users and items, respectively, where $|\mathcal{U}|$, $|\mathcal{I}|$ denote the number of users and items. Assuming that there exists a chain of sequential behavior $\mathcal{B} = \{b_1, b_2, \ldots, b_K\}$ (e.g., *page-view* → *add to cart* → *purchase*), and the matrix of user interaction with an item is denoted as $\mathrm{Y} = \{\mathbf{Y}^1, \mathbf{Y}^2, \ldots, \mathbf{Y}^K\}$, where K is the total number of behavior types, $\mathbf{Y}^K$ denotes the target behavior matrix, and $\{\mathbf{Y}^1, \mathbf{Y}^2, \ldots, \mathbf{Y}^{K-1}\}$ is the auxiliary behavior matrix. $y_{ui}^k \in \{0, 1\}$ indicates whether or not user u interacts with the item i under the behavior b_k. N_u^k or N_i^k denotes the set of items interacted with by user u or the set of users who interacted with item i under $b_k \in \mathcal{B}$.

Directed behavior path graph. Let $\mathcal{G}_{all} = (\mathcal{B}, \mathcal{E})$ represents the directed behavior path graph between users and items, $\mathcal{E} = \{(b_{k'}, b_k) | 0 \le k' < k \le K\}$ represents the directed behavior edges, where the nodes in the graph represent behaviors, and the directed edges indicate the transition relationships between behaviors.

Target behavior and auxiliary behavior. In multi-behavior recommendation tasks, purchase is the strongest signal that directly impacts the total value of an item and is regarded as the target behavior b_K. We use $\{b_1, b_2, \ldots, b_{K-1}\}$ to represent auxiliary behaviors, which provide additional observed interactions to assist in predicting the target behavior.

Input. Given the multiplex user-item interactions $\{\mathbf{Y}^1, \mathbf{Y}^2, \ldots, \mathbf{Y}^K\}$ between the user set $\mathcal{U}$ and the item set $\mathcal{I}$.

Output. Predicting a ranking score $\widehat{y}_{ui}^K$ denotes the likelihood of user u purchasing item i under the target behavior K. The candidate items are ranked in descending order to generate the Top-K item recommendation list.

3.2 The Overall Framework

Fig. 2 illustrates the proposed framework architecture. The entire process can be divided into the following four steps:

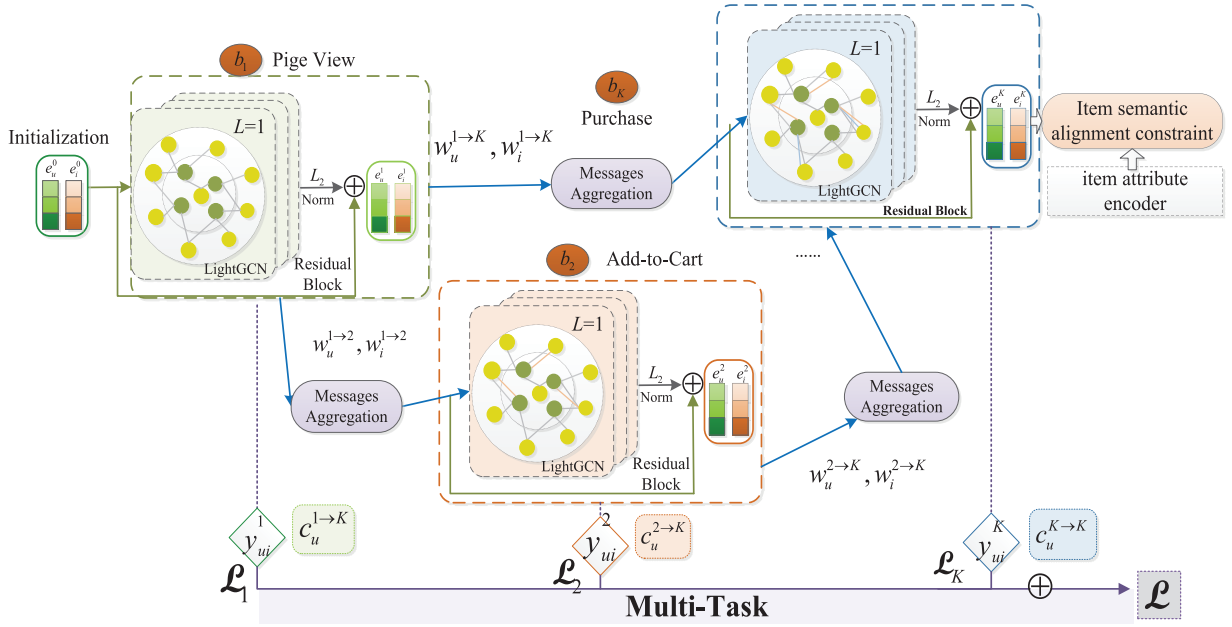

Figure 2: Overall framework of DBMR.

1. Construction of the directed behavior path graph. In this paper, a directed behavior path graph for users and items is constructed through diversified behavior chains, which includes all observed behavior chains and initializes the edge weights in the graph.

2. Message propagation in the user-item interaction graph. We construct a user-item interaction graph for each behavior in the directed behavior path graph. The LightGCN model with residual design is applied to fuse the information of adjacent nodes for each node in the user-item interaction graph into the embeddings, to learn the feature embeddings of users and items under each behavior.

3. Message aggregation in the directed behavior path graph. We propose a graph attention message aggregator, which effectively aggregates the user and item feature embeddings under different behaviors based on the varying behavior transition strengths. The aim is to weight and aggregate the features learned from previous behaviors into the feature learning of the next behavior.

4. Joint optimization. In the model optimization, we adopt personalized multitask learning, which determines the contribution of each auxiliary behavior to the target behavior prediction by measuring the similarity of interactions across different behaviors. At the same time, an auxiliary loss function is introduced to measure the difference between the item attribute embeddings obtained based on behavioral paths and the item attribute embeddings learned through the Attribute Feature Encoder, as Fig. 3. This provides the model with multidimensional information.

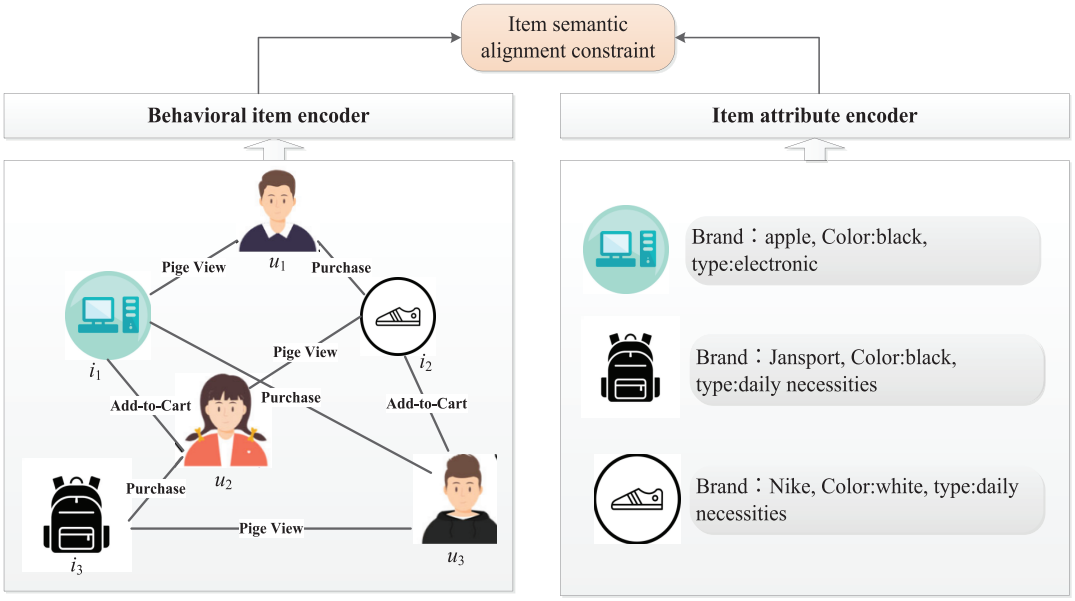

Figure 3: Framework diagram of item semantic alignment constraints.

3.3 Construction of the Directed Behavior Path Graph

Our goal is to make recommendations for users based on diversified behavior chains. To achieve this, we construct a unified directed behavior path graph to model the problem under investigation.

We follow real-world recommendation scenarios, where the behavior chain begins with page-view behavior and supports jump behaviors and nonlinear paths. Base on behavior chains $\mathcal{B} = \{b_1, b_2, \ldots, b_K\}$, we construct a directed behavior path graph $\mathcal{G} = (\mathcal{B}, \mathcal{E})$, where b_K denotes behavior. $\mathcal{E} = \{(b_{k'}, b_k)|0 \leq k' < k \leq K\}$ denotes a directed behavior edge, which represents the behavior transformation relationship. We construct a directed behavior path graph for each user and each item $\mathcal{G}_u = (\mathcal{B}, \mathcal{E}_u)$, $\mathcal{G}_i = (\mathcal{B}, \mathcal{E}_i)$, which has the same graph structure, but each directed edge has a different weight. $w_u^{k' \to k}$, $w_i^{k' \to k}$ denotes the strength of behavior transition from behavior $b_{k'}$ to behavior b_k for user u or item i. In a multi-behavior scenario with K behaviors, the directed behavior path graph G is divided into K user-item interaction graphs under different behaviors.

Let $\mathbf{G}_u^k = \{\mathbf{V}_u^k, \mathbf{E}_u^k\}$, $\mathbf{G}_i^k = \{\mathbf{V}_i^k, \mathbf{E}_i^k\}$ denote user-item interaction graph under the k-th behavior. Using the user-item graph $\mathbf{G}_u^{pv} = \{\mathbf{V}_u^{pv}, \mathbf{E}_u^{pv}\}$ under page view behavior as an example. Let $\mathbf{V}_u^{pv} \in \{\mathbf{U} \cup \mathbf{I}\}$ denote the set of nodes in the graph, which represents the set of items that users interact with during browsing behavior. Let $\mathbf{E}_u^{pv}$ denote the set of edges in the graph, which represents that user u has purchased item i. In the user-item interaction graph, the adjacent node information of each node is fused into the embedding through the user-item propagation layer, to enable the learning of user and item feature embeddings for each type of behavior.

3.4 User-Item Interaction Graph Message Propagation

3.4.1 Embedding Initialization

Following the general approach used in existing recommendation methods, we initialize the $u \in \mathbf{U}$ and $i \in \mathbf{I}$ embedding vector as $e_u^0 \in \mathbb{R}^d$, $e_i^0 \in \mathbb{R}^d$, where d represents embedding size. Let $\mathbf{P} \in \mathbb{R}^{U \times d}$, $\mathbf{Q} \in \mathbb{R}^{I \times d}$ represent the embedding matrices of users and items, respectively. Each user or item has a unique ID represented by a one-hot vector, where ID^U, ID^I are the one-hot vector matrix for all users and items. It is then transformed into low-dimensional embeddings through a shared embedding layer with $\mathbf{P} = \{\mathrm{p}_{u_1}^0, \mathrm{p}_{u_2}^0, \ldots, \mathrm{p}_{u_{|\mathcal{U}|}}^0\}$ and $\mathbf{Q} = \{q_{i_1}^0, q_{i_2}^0, \ldots, q_{i_{|\mathcal{I}|}}^0\}$. The embeddings of user u and item i are initialized as:

$$e_u^0 = \mathbf{P}^\top \cdot \mathrm{ID}_u^U, e_i^0 = \mathrm{Q}^\top \cdot \mathrm{ID}_i^I, \tag{1}$$

where ID_u^U, ID_i^I denote the one-hot vectors of users and items, respectively.

3.4.2 User-Item Interaction Graph Encoder

Inspired by previous works [31], we have designed a user-item interaction graph encoder based on LightGCN with a residual design, where each LightGCN corresponds to a specific behavior, as shown in Fig. 4. This residual design uses residual connections to directly propagate the node representations from the previous layer to the next layer, thereby effectively preserving the node's own features and mitigating the over-smoothing issue in networks. The output representations of all layers in the network maintain the same dimension d as the initial embedding, ensuring that representations from different layers can be directly fused. In this module, the residual fusion weights adopt a non-parametric and equal-weight setting, meaning the contribution weight of each layer is fixed at 1, enabling simple and efficient inter-layer information transfer. This constitutes a non-parametric residual connection. This constitutes a non-parametric residual connection. Through an iterative aggregation and update mechanism, it captures multi-order connections in the user-item interaction graph under different behaviors, thereby learning the user and item embeddings for each behavior. Given the representation e_u^k, e_i^k for user u and item i under behavior b_k, the aggregation and update of the user-item interaction graph can be formulated as:

$$e_u^{k,(l+1)} = \sum_{i \in N_u^k} \frac{1}{\sqrt{|N_u^k|}\sqrt{|N_i^k|}} e_i^{k,(l)}, \tag{2}$$

$$e_i^{k,(l+1)} = \sum_{u \in N_i^k} \frac{1}{\sqrt{|N_i^k|}\sqrt{|N_u^k|}} e_u^{k,(l)}, \tag{3}$$

where $e_u^{k,(l+1)}$, $e_i^{k,(l+1)}$ represent the user u and item i embeddings after $l+1$ layers of propagation and update under behavior K. Let N_u^k, N_i^k represent the set of items interacted with by user u and the set of users who have interacted with item i.

After L layers of propagation, the user's embedding is represented as $\{e_u^{(b,0)}, e_u^{(b,1)}, \ldots, e_u^{(b,L)}\}$ and item's embedding is represented as $\{e_i^{(b,0)}, e_i^{(b,1)}, \ldots, e_i^{(b,L)}\}$. We aggregate the user and item representations under behavior K as:

$$e_u^k = \sum_{l=0}^{L} e_u^{(k,l)}, e_i^k = \sum_{l=0}^{L} e_i^{(k,l)}, \tag{4}$$

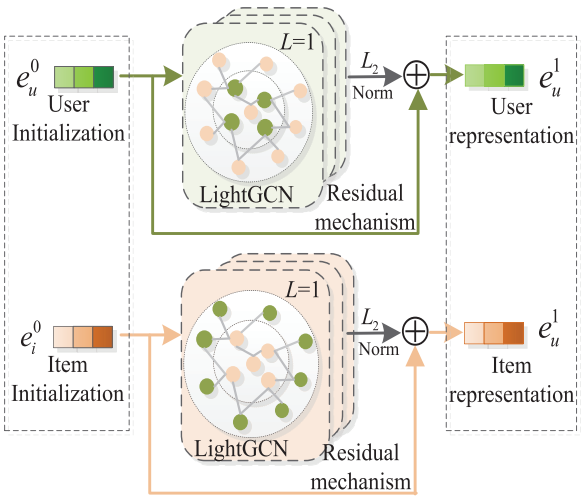

Figure 4: User-item interaction diagram message propagation.

After the feature transformation operation, e_u^k, e_i^k, as learned from the behavior b_k, are used as the input embedding vectors for the LightGCN of the next behavior. Importantly, the representations learned from the previous LightGCN are used as the input features for the users and items in the next LightGCN. Importantly, the representation of the behavior b_k is used as the initial user and item information for the LightGCN encoder under behavior b_k, which can be expressed as:

$$e_u^{k,(0)} = e_u^{k'}, e_i^{k,(0)} = e_i^{k'}. \tag{5}$$

3.5 Directed Behavior Path Graph Message Aggregation

We propose a graph attention message aggregator that effectively aggregates user and item representations under different behaviors with the varying transition strengths of behaviors. Features learned from previous behaviors are weighted and aggregated into the feature learning of the next behavior, updating user preferences and item characteristics through the behavior chain.

3.5.1 Initialization of Behavior Transition Strength Matrix

Different types of behaviors reveal varying degrees of user preferences. In a behavior chain, the subsequent behavior typically exhibits stronger signals or more accurate user preferences than the preceding behavior. However, the contributions of different auxiliary behaviors to the target behavior vary. Ignoring these differences may lead to the loss of the diverse information conveyed by the different behaviors. Therefore, we introduce the behavior transition strength matrix from behavior to behavior. Different types of behaviors reveal varying degrees of user preferences. In a behavior chain, the subsequent behavior typically exhibits stronger signals or more accurate user preferences than the preceding behavior. However, the contributions of different auxiliary behaviors to the target behavior vary. Ignoring these differences may lead to the loss of the diverse information conveyed by the different behaviors. Therefore, we introduce the behavior transition strength matrix from behavior $b_{k'}$ to behavior b_k.

Behavior Transition Strength Matrix. Let $\mathbf{W}_U^k \in \mathbb{R}^{U \times k}$ be the behavior transition strength Matrix, which represents the user-adaptive distribution of all auxiliary behaviors $\{b_1, b_2, \ldots, b_{k-1}\}$. $\mathbf{W}_U^{k' \to k} \in \mathbb{R}^U$ represents the behavior transition strength matrix from behavior $b_{k'}$ to behavior b_k for all users, which $0 \leq k' \leq k-1$. In the directed behavior path graph, a specific transition strength weight is assigned to each

directed edge. We initialize the transition strength weights for each graph and update them through learning in subsequent modules.

Inspired by the Markov process, we only consider the nearest preceding behavior in each behavior path, assuming that it has the most direct influence on the terminating behavior. We investigated the correlation from the given starting behavior $b_{k'}$ to the terminating behavior b_k, comprehensively considering all paths leading to the terminating behavior and fully accounting for the influence of preceding behaviors in its formation. Taking the user's directed behavior path graph as an example, we calculate the probability of items that appeared only in the starting behavior $b_{k'}$ appearing in the terminating behavior b_k. First, we remove the items from the terminating behavior b_k that appeared in the behaviors between $b_{k'}$ and b_k. Then, we take the intersection with $N_u^{k'}$ to get the result, which represents the set of items directly transitioning from behavior $b_{k'}$ to b_k in the user's behavior path. Finally, we use the posterior probability of the terminating behavior b_k to normalize its correlation with all possible starting behaviors, obtaining the behavior transition strength from behavior resents the set of items directly transitioning from behavior $b_{k'}$ to b_k for the user and item. This is formally represented as follows:

$$w_u^{k' \to k} = \frac{\left| \left(N_u^k - \cup_{j=kv+1}^{k-1} N_u^j \right) \cap N_u^{k'} \right|}{|N_u^k|}, \tag{6}$$

$$w_i^{k' \to k} = \frac{\left| \left(N_i^k - \cup_{j=k'+1}^{k-1} N_i^j \right) \cap N_i^{k'} \right|}{|N_i^k|}, \tag{7}$$

where N_u^k, N_i^k represent the set of items interacted with by user u and the set of users interacting with item i under behavior $b_k \in \mathcal{B}$. To mitigate scaling differences, we employ weight normalization to account for message variations across different behaviors, using the Softmax function to normalize the behavior distribution:

$$\widetilde{w}_u^{k' \to k} = \frac{w_u^{k' \to k}}{\sum_{k'=0}^{k-1} w_u^{k' \to k}}, \tag{8}$$

$$\widetilde{w}_i^{k' \to k} = \frac{w_i^{k' \to k}}{\sum_{k'=0}^{k-1} w_i^{k' \to k}}. \tag{9}$$

3.5.2 Graph Attention Message Aggregation Encoder

We propose a graph attention-based information aggregation encoder, which effectively aggregates user and item feature embeddings across different behaviors based on varying behavior transition strengths. It updates user preferences and item characteristics through the behavior chain. To preserve the features of previous behaviors, our model employs a residual operation to connect the feature embeddings learned from prior behaviors with those learned from current behavior. This method connects the feature embeddings learned from previous behaviors with those learned from the current behavior, enabling the model to effectively learn high-order interaction patterns while retaining crucial information from the original behavioral features. All embeddings maintain the same feature dimension d, including the embedding of the current behavior b_k, the embedding of its direct predecessor behavior b_{k-1}, and the embeddings corresponding to all historical behaviors. This ensures that features from each stage can be directly fused via weighted aggregation. In this module, the residual fusion weights are set as non-learnable parameters, with the weight for the residual term corresponding to the direct predecessor behavior b_{k-1} fixed at 1. This enables

our model to benefit from higher-order GCN operations while retaining the original behavior features. We update user and item embeddings using a graph attention information aggregation encoder:

$$\widetilde{e}_u^k = e_u^{k-1} + \sum_{k'=0}^{k-1} \widetilde{w}_u^{k' \to k} e_u^{k'}, \tag{10}$$

$$\widetilde{e}_i^k = e_i^{k-1} + \sum_{k'=0}^{k-1} \widetilde{w}_i^{k' \to k} e_i^{k'}, \tag{11}$$

where $\widetilde{e}_u^k$, $\widetilde{e}_i^k$ represent the user preference representation and item characteristic representation under behavior b_k, respectively. Similarly, $e_u^{k'}$, $e_i^{k'}$ represent the user preference representation and item characteristic representation under behavior $b_{k'}$.

3.6 Joint Optimization

Through the above module, we learn the embedding representations of users and items for each type of behavior. For each behavior, we estimate the user's preference for the candidate items by calculating the dot product, which can be computed as:

$$\widehat{y}_{ui}^k = \widetilde{e}_u^{k\top} \widetilde{e}_i^k, \tag{12}$$

where $\hat{y}_{u,i} \in [0,1]$ denotes the relevance score between the user and the candidate item i. In the Top-K recommendation task, the K candidate items with the highest interaction probabilities for user u will be selected to form the recommendation list.

3.6.1 Multi-Task Training

In the model optimization process, we introduce the Multi-Task Learning (MTL) framework [32,33] to quantify the contribution of each auxiliary task to the target task prediction, enabling collaborative optimization across different tasks. Therefore, to ensure that the target task can effectively benefit from the predictions of auxiliary interactions, it is essential to accurately measure the correlation between the target behavior and auxiliary behaviors. We calculate the item similarity under the target and auxiliary behaviors as a metric for evaluating the correlation between the two behaviors. We calculate the correlation score $c_u^{k \to K}$ from the auxiliary behavior $\{b_1, b_2, \ldots, b_{K-1}\}$ to the target behavior b_K using the Cosine similarity for user u, which can be defined as:

$$c_u^{k \to K} = \frac{|N_u^k \cap N_u^K|}{\sqrt{|N_u^k|} \cdot \sqrt{|N_u^K|}}, k = 1, 2, \ldots, K, \tag{13}$$

where N_u^k is the set of items interacted with by user u under behavior b_k. For a specific user, a higher correlation score indicates a stronger interaction between the given auxiliary behavior and the target behavior, suggesting that the auxiliary behavior has a significant impact on the prediction of the target behavior.

Similar to other ranking-based recommendation approaches, a pairwise learning strategy is employed for model optimization. In the multi-behavior dataset, Bayesian Personalized Ranking (BPR) is used as the optimization criterion during the training phase. The BPR loss function is widely used in recommendation systems [10], and by emphasizing the relative order of observed and unobserved user-item interactions, it guides the model to capture user preferences and item characteristics more accurately. Based on this, a loss

function is constructed to ensure that the model assigns higher predicted scores to observed interactions. The loss function is formulated as follows:

$$\mathcal{L}_k = \sum_{(u,i,j)\in O^k} -c_u^{k\to K}(\ln \sigma(\widehat{y}_{ui}^k - \widehat{y}_{uj}^k)), \tag{14}$$

where $O_k = \{(u,i,j)|(u,i) \in \mathrm{Y}^{k,+}, (u,j) \in \mathrm{Y}^{k,-}\}$ is the training set for behavior b_k. $\mathrm{Y}^{k,+}$ ($\mathrm{Y}^{k,-}$) represent the observed (or unobserved) samples in the target behavior. σ denotes the sigmoid function.

3.6.2 Item Semantic Alignment Constraint

To achieve the joint modeling of behavioral patterns and item semantics, we propose an item semantic alignment constraint. This constraint aligns the feature representations from two distinct perspectives within a latent space by minimizing the discrepancy between item features learned from behavioral path graphs and item attribute features obtained via an attribute encoder, thereby enhancing the representational capacity of item features. First, we use the pre-trained language model Recformer [34] to learn item-attribute embeddings. This model transforms all textual attribute information for an item (e.g., title, color, brand) into key-value pairs, allowing different items to have different sets of attributes. Subsequently, all key-value pairs of an item are flattened into an item "sentence". Finally, the pre-trained language model is utilized to derive the feature representation of this item sentence. Formally, given an item i with n attributes, its key-value pairs $\{(x_1^i, v_1^i), (x_2^i, v_2^i), \cdots, (x_n^i, v_n^i)\}$ are flattened into a sequence $\{x_1^i, v_1^i, x_2^i, v_2^i, \cdots, x_n^i, v_n^i\}$. Prepending the special [CLS] token, the input is fed into the Recformer model:

$$\widetilde{e}_i = \mathrm{Recformer}(\mathrm{Flatten}(\{(x_1^i, v_1^i), (x_2^i, v_2^i), \ldots (x_n^i, v_n^i)\})) \tag{15}$$

where $(x_1^i, v_1^i), (x_2^i, v_2^i), \ldots (x_n^i, v_n^i)$ denotes the n attribute key-value pairs of item i. Flatten$(\cdot)$ represents flattening the key-value pair sequence into a continuous text sequence and adding a special [CLS] token at the beginning. Recformer$(\cdot)$ denotes a pretrained language model, which takes the flattened text sequence as input and outputs the hidden state corresponding to the [CLS] token as the attribute semantic feature $\widetilde{e}_i$ of item i.

To establish synergy between behavioral patterns and item semantics, the proposed alignment constraint reduces the distance in the vector space between the behavior-learned item representation and the semantically-learned attribute representation. Specifically, for item i, its behavior-based feature $\widetilde{e}_i^K$ and its attribute feature $\widetilde{e}_i$ form a positive pair, while $\widetilde{e}_i^K$ and the attribute features of other items in the batch form negative pairs. The contrastive loss is defined as:

$$\mathcal{L}_{\mathrm{contra}} = -\frac{1}{|\mathcal{I}_{\mathrm{batch}}|} \sum_{i\in\mathcal{I}_{\mathrm{batch}}} \log \frac{\exp(\mathrm{sim}(\widetilde{e}_i^K, \widetilde{e}_i)/\tau)}{\sum_{j\in\mathcal{I}_{\mathrm{batch}}} \exp(\mathrm{sim}(\widetilde{e}_i^K, \widetilde{e}_j)/\tau)}, \tag{16}$$

where $\mathcal{I}_{\mathrm{batch}}$ is the set of items in the current training batch, sim$(\cdot)$ denotes the cosine similarity function, and $\tau > 0$ is the temperature parameter.

This contrastive loss effectively promotes the deep integration of behavioral patterns and item semantics by maximizing the agreement between the behavioral-view and semantic-view representations of the same item while minimizing the similarity between representations of different items, thereby enhancing the discriminative power of the learned item features. Finally, the objective function of the DBMR model proposed in this paper can be expressed as:

$$\mathcal{L} = \mathcal{L}_k + \alpha \sum_{k=0}^{K-1} \mathcal{L}_k + \lambda \mathcal{L}_{\text{contra}} + \beta ||\Theta||_2^2, \tag{17}$$

where α is the discounting parameter between the auxiliary task and the target task, λ is the coefficient of the semantic alignment constraint function for the item, and β controls the L_2-regularization to prevent overfitting. $\Theta = \{\mathbf{P}, \mathbf{Q}, \mathbf{W}_U, \mathbf{W}_I\}$ is the representation of all trainable parameters. The detailed training process of DBMR is shown in the Algorithm 1.

Algorithm 1: DBMR

Input: Dataset $\mathcal{D}_{\text{train}} = \{\langle u, i, r_{u,i}^k \rangle \mid u \in \mathcal{U}, i \in \mathcal{I}, k \in \mathcal{B}\}$, User Set $\mathcal{U}$, Item Set $\mathcal{I}$, Behavior Set $\mathcal{B}$, Embedding Dimension d, Number of Graph Convolution Layers L, Hyperparameters α, β, Batch Size$|\mathcal{B}|$, Learning Rate η

Output: Model Parameters Θ

Initialize all model parameters Θ;

Construct a directed behavior path graph $\mathcal{G}_{Direct} = (\mathcal{B}, \mathcal{E})$;

repeat

for *Each mini-batch* $\mathcal{B} \subset \mathcal{D}_{train}$ **do**

for *Each training sample* $\langle u, i, y_{u,i}^k \rangle \in \mathcal{B}$ **do**

for each i in $\mathcal{I}$ **do**

Using Formula (1), calculate the initial embedding representation of the item e_i^0;

for each u in $\mathcal{U}$ **do**

Using Formula (1), calculate the initial embedding representation of the user e_u^0;

for *Each training* $k \in \mathcal{B}$ **do**

Construct the user-item interaction graph $\mathcal{G}^k$ for behavior k;

Calculate the user feature representation e_u^k and item feature representation e_i^k under behavior k using Eqs. (2)–(4);

Using Eqs. (8) and (9), calculate the user behavior transition strength $\widetilde{w}_u^{k' \to k}$ and item behavior transition strength $\widetilde{w}_i^{k' \to k}$;

Using Eqs. (10) and (11), calculate the final user feature representation $\widetilde{e}_u^k$ and item feature representation $\widetilde{e}_i^k$;

Using Eq. (12), calculate the predicted rating $\hat{y}_{u,i}$ of user u for item i under the target behavior;

Using Eq. (14), calculate the multi-task loss $\mathcal{L}_{\text{multi}}$;

Using Eq. (16), calculate the semantic alignment loss $\mathcal{L}_{\text{contra}}$;

Using Eq. (17), Compute the overall loss $\mathcal{L}$ for the current batch ;

Minimize the loss $\mathcal{L}$ using the Adam optimizer via gradient descent to update the model parameters Θ;

until *Network Convergence*;

return Θ;

4 Algorithm Complexity Analysis

We conduct an algorithm complexity analysis from both temporal and spatial perspectives.

4.1 Time Complexity

The time complexity primarily consists of four components: the construction of the directed behavior path graph, the propagation in the user-item interaction graph, the aggregation across behavior paths, and

the semantic alignment and optimization. First, the construction of the directed behavior path graph is based on the historical interaction behaviors between users and items, with a complexity of $O(K^2 \cdot (|\mathcal{U}| + |\mathcal{I}|) \cdot \overline{m})$, where K represents the number of behavior types, and $\overline{m}$ denotes the average number of items interacted with per user. Next, in the propagation of the user-item interaction graph, the complexity of each LightGCN convolutional layer is $O(|\varepsilon^k|d)$, and the total complexity for L layers is $O(L \cdot |\varepsilon|_{total} \cdot d)$, where $|\varepsilon|_{total} = \sum_{k=1}^{K} |\varepsilon^k|$ represents the total number of user-item interaction edges across all behaviors. Following this, the cross-behavior path aggregation module performs a weighted summation, with a complexity of $O(K \cdot d \cdot (|\mathcal{U}| + |\mathcal{I}|))$. Finally, the in-batch contrastive loss L_{contra} in the semantic alignment and optimization module has a complexity of $O(B^2 \cdot d)$, where B denotes the batch size.

4.2 Space Complexity

The space complexity is primarily analyzed in four aspects: user/item ID embeddings, adjacency matrices of K sparse interaction graphs, item semantic embeddings, and behavior transition weight matrices. The storage overhead for user/item ID embeddings is $O((|\mathcal{U}| + |\mathcal{I}|) \cdot d)$, the storage overhead for adjacency matrices of K sparse interaction graphs is $O(|\varepsilon|_{total})$, the storage overhead for item semantic embeddings is $O(|\mathcal{I}| \cdot d)$, and the storage overhead for behavior transition weight matrices is $O(K^2 \cdot (|\mathcal{U}| + |\mathcal{I}|))$.

5 Experiments

5.1 Experiment Settings

5.1.1 Datasets

To evaluate the performance of DBMR, we conducted experiments on two real-world publicly available datasets: Tmall and Taobao. The statistics of these two datasets are summarized in Table 1.

Table 1: Statistical Results of the Datasets.

Dataset	# Users	# Items	# Interactions	#Page View	#Add-to-Cart	# Purchase
Tmall	15,449	11,953	1.2×10^6	873,954	195,476	104,329
Taobao	48,749	39,493	2.0×10^6	1,548,162	193,747	211,022

Tmall[1]: The dataset encompasses multiple types of user–item interactions, including page view, favoriting, adding to cart, and purchasing. It also provides rich auxiliary information, such as user and item attributes, interaction timestamps, and multimodal content composed of item titles and descriptive images. Specifically, user attributes include age and gender, while item attributes comprise category, brand, and seller information.

Taobao[2]: This dataset is specifically designed to collect and organize user shopping behaviors on Taobao. It generally includes users' page-view history, purchase records, search behavior, review feedback, and item attributes.

5.1.2 Evaluation

To ensure fairness and facilitate comparison with existing methods, we merged duplicate user-item interaction data and filtered out users and items with fewer than 5 purchase interactions to ensure the

[1] https://www.tmall.com
[2] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649

representativeness and validity of the data. Subsequently, the last purchase record of each user was taken as the test set, while the remaining records were used as the training set. In the performance evaluation phase, we employed two commonly used Top-K recommendation evaluation metrics: Normalized Discounted Cumulative Gain (N@K) and Hit ratio (H@K), where K represents the length of the recommendation list and K = 10, 20, 50, 80.

5.1.3 Baselines

To thoroughly evaluate the recommendation performance of DBMR, this study selected two types of baseline algorithms (single-behavior recommendation models and multi-behavior recommendation models) for comparative experiments.

(1) Single-Behavior Recommendation Model

NeuMF [35]: It is a typical deep learning-based recommendation algorithm that enhances nonlinear matrix factorization through a multi-layer perceptron module and a generalized matrix factorization module, allowing it to simultaneously capture both low-dimensional and high-dimensional user-item interaction signals.

MF-BPR [10]: It is a widely used matrix factorization model with BPR as its objective function, optimized using Bayesian Personalized Ranking (BPR) loss for matrix factorization, applied to Top-K recommendation tasks.

LightGCN [3]: It is a lightweight graph convolutional network recommendation model that eliminates feature transformations and nonlinear activations from NGCF, making it simpler and more suitable for recommendation tasks.

(2) Multi-Behavior Recommendation Model

NMTR [23]: This approach integrates Neural Collaborative Filtering (NCF) with multi-task learning to model the dependencies of various types of user interaction behaviors, mining user preferences from multi-behavior data for recommendation.

MB-GMN [36]: It is a recommendation method capable of extracting effective representations of users and items from complex multi-behavior data. This model combines meta-learning with graph neural networks, utilizing a graph meta-network to capture meta-knowledge about users, items, and behavior categories, thereby capturing behavior-based user and item feature information for accurate recommendations.

S-MBRec [37]: This model learns the importance of different behaviors in a supervised manner, while capturing the commonalities and differences in user behaviors, thereby improving the accuracy of recommendations.

CRGCN [24]: This method utilizes a cascaded GCN model with residual connections to learn user preferences and optimizes the model under a multi-task recommendation framework.

MB-CGCN [25]: This method effectively models the dependencies of behaviors within a sequential behavior chain, using user and item embeddings learned from one behavior as input features for the embedding learning process of the next behavior. Additionally, a feature transformation module is designed before each behavior's information propagation stage in the model, effectively filtering and transforming noisy information.

DA-GCN [26]: This method utilizes the partial order relationship in a monotonic behavior chain and extends it to a personalized directed acyclic behavior graph to leverage behavior dependencies for recommendation.

In the aforementioned multi-behavior recommendation baselines, NMTR, CRCCN, and MB-CGCN are categorized as cascaded models, primarily relying on the determined behavior chains. In contrast, MB-GMN, S-MBRec, and DA-GCN are classified as parallel models, typically using higher-order operations to process the semantic and contextual information of each behavior separately, thereby forming complex and diverse model architectures.

5.1.4 Hyper-Parameter Settings

For all models, unless explicitly stated in the literature, this experiment uses the Adam optimizer and stochastic gradient descent strategy for model optimization and training, with the Adam optimizer's learning rate set to $1e^{-3}$. For the comparison methods, we first initialize the baseline algorithms with the parameters specified in the literature, and then tune the parameters based on the dataset and optimization strategy. To ensure a fair comparison, the embedding dimensions of both the models and the baseline comparison methods are set to 64. For the user-item interaction graph, we explore the number of Light-GCN layers for each behavior within the set 1, 2, 3, 4. For multitasking, the goal reference [26] recommend varying the trade-off hyperparameter between the target task and auxiliary tasks over the range 0.2 to 3.0, with a step size of 0.2, and setting the regularization coefficient to 0.001.

5.2 Performance Comparison

Experiments were conducted on two public datasets (Tmall and Taobao) to compare the performance of the proposed DBMR method with baseline methods. Tables 2 and 3 present the performance comparison results between DBMR and baseline methods for Top-*K* recommendation, with the recommendation list length set to 10, 20, 50, and 80, respectively. For performance comparison purposes, this section adopts the Impro metric to quantify the improvement margin of DBMR over comparative methods in terms of H@K and N@K. The best and second-best results on the Tmall and Taobao datasets are marked in bold and underlined in Tables 2 and 3. Based on the analysis of experimental results, the following conclusions can be drawn:

(1) The proposed DBMR method demonstrates performance improvements on both the Tmall and Taobao datasets, surpassing the current best baseline, DA-GCN. Specifically, as shown in Table 2, on the Tmall dataset, DBMR achieves maximum improvements of 17.46% and 16.5% in H@K and N@K metrics, respectively, compared to the optimal baseline DA-GCN. As presented in Table 3, on the Taobao dataset, it attains maximum gains of 14.13% in H@K and 5.39% in N@K. In contrast, DA-GCN also employs directed graph modeling and achieves the second-best performance. Although this method can leverage directed behavior graphs to learn basic behavioral dependencies, its limitations in modeling complex user decision processes are due to insufficient consideration of the topological structure of behavioral paths and their interaction with item attributes, resulting in marginally inferior performance compared to the proposed DBMR method.

(2) Multi-behavior models consistently outperform single-behavior models, a phenomenon that sufficiently demonstrates how the incorporation of auxiliary behaviors helps enrich semantic representations and exerts a positive influence on predicting the target behavior. In Table 2, multi-behavior models achieve higher H@K and N@K values than single-behavior models on the Tmall dataset. Similarly, as shown in Table 3, substantial performance improvements are also observed on the Taobao dataset. These results validate the need to incorporate multi-behavior data and underscore the critical role of auxiliary behaviors in improving the accuracy of target behavior recommendations.

(3) Multi-behavior models employing graph convolutional networks as semantic extractors (such as CRGCN, MB-CGCN, and the proposed DBMR) demonstrate superior performance compared to models

solely based on multilayer perceptrons (e.g., NMTR). This result can be attributed to the inherent capability of graph convolutional networks in effectively capturing high-order neighborhood information, thereby substantially enhancing performance in recommendation tasks. Specifically, among single-behavior models, LightGCN significantly outperforms MF-BPR, achieving a 87.33% improvement, further validating the notable advantage of GCN models in leveraging high-order neighbor information during the learning of user and item embeddings. Owing to its exceptional ability in learning node representations within graph structures, GCN has effectively propelled the continuous development of GCN-based multi-behavior recommendation models. Consequently, the DBMR method proposed in this chapter fully incorporates the LightGCN framework.

(4) The proposed DBMR method demonstrates significantly superior performance compared to most existing cascade models, including CRGCN with residual connection structures and MB-CGCN incorporating feature transformation mechanisms. These results sufficiently validate the critical role of explicit behavioral path modeling in enhancing recommendation performance, while emphasizing the necessity and irreplaceability of considering diversity and completeness during the modeling process. Furthermore, by incorporating semantic constraints from item attributes into behavioral paths, DBMR obtains more discriminative and robust item representations, ultimately systematically improving both the recommendation effectiveness and the model's generalization capability.

Table 2: Comparison of the Recommendation Performance of Our Method and Baseline Algorithms in terms of H@K and N@K on the Tmall Dataset.

Type	Model	H@10	H@20	H@50	H@80	N@10	N@20	N@50	N@80
Single-Behavior	NeuMF	0.012	0.019	0.0382	0.051	0.0056	0.0074	0.0111	0.0133
	MF-BPR	0.023	0.0316	0.0434	0.0541	0.0124	0.0144	0.0166	0.0183
	LightGCN	0.0393	0.0538	0.0813	0.0984	0.0209	0.0243	0.0295	0.0322
Multi-Behavior	NMTR	0.0195	0.0361	0.0773	0.1133	0.0086	0.0127	0.0209	0.0268
	MB-GMN	0.0481	0.0677	0.1003	0.1249	0.0317	0.0364	0.0418	0.0466
	S-MBRec	0.0694	0.1009	0.1553	0.1901	0.0362	0.0438	0.0544	0.0601
	CRGCN	0.0945	0.1386	0.2197	0.2739	0.0514	0.0625	0.0784	0.0875
	MB-CGCN	0.0984	0.1487	0.2581	0.3002	0.0558	0.088	0.0971	0.1134
	DA-GCN	0.1451	0.193	0.2764	0.3256	0.0897	0.0952	0.1117	0.1192
	DBMR	**0.1666**	**0.2267**	**0.2857**	**0.3339**	**0.1045**	**0.1118**	**0.1234**	**0.1298**
Impro		14.82%	17.46%	3.36%	2.55%	16.5%	12.24%	10.47%	8.89%

Table 3: Comparison of the Recommendation Performance of Our Method and Baseline Algorithms in terms of H@K and N@K on the Taobao Dataset.

Type	Model	H@10	H@20	H@50	H@80	N@10	N@20	N@50	N@80
Single-Behavior	NeuMF	0.0152	0.0216	0.0332	0.0411	0.0088	0.0104	0.0127	0.014
	MF-BPR	0.0157	0.0217	0.0309	0.0373	0.0089	0.0105	0.0123	0.0134
	LightGCN	0.0379	0.0498	0.071	0.0818	0.0219	0.0249	0.0291	0.0309
	NMTR	0.0128	0.0275	0.0656	0.0943	0.0053	0.0089	0.0164	0.0211
	MB-GMN	0.0438	0.0621	0.0956	0.1559	0.0326	0.0365	0.0421	0.0476

(Continued)

Table 3 (continued)

Type	Model	H@10	H@20	H@50	H@80	N@10	N@20	N@50	N@80
Multi-Behavior	S-MBRec	0.0629	0.0903	0.1269	0.1463	0.0307	0.0376	0.045	0.0482
	CRGCN	0.0985	0.1361	0.1987	0.238	0.057	0.0665	0.0788	0.0854
	MB-CGCN	0.1233	0.1624	0.2347	0.2603	0.0677	0.0829	0.0964	0.1068
	DA-GCN	0.1588	0.1971	0.2527	0.2928	0.1007	0.1104	0.1224	0.1283
	DBMR	**0.1596**	**0.2032**	**0.2884**	**0.3308**	**0.1009**	**0.1121**	**0.129**	**0.136**
Impro		0.50%	3.09%	14.13%	12.98%	0.20%	1.54%	5.39%	6.0%

5.3 Ablation Study

To validate the effectiveness of the designed attention mechanism module and the item semantic alignment constraint module in the DBMR method, ablation studies were conducted on the corresponding components.

5.3.1 Attention Mechanism Module

The proposed DBMR employs attention mechanisms in two distinct modules: behavioral transition intensities for message aggregation in directed behavioral path graphs and behavioral correlation for multi-task learning. To analyze the impact of different attention mechanisms on recommendation performance, this section compares DBMR with the following two method variants under specific configurations:

w/o BTS: This variant removes the Behavior Transition Strengths (BTS) module from the DBMR method, which learns contributions of auxiliary behaviors. By setting the behavioral transition strength w in Eqs. (8) and (9) to the same value, this variant treats all behaviors' influence on the target equally during modeling.

w/o BRE: This variant removes the Behavioral Relevance (BRE) module from the multi-task learning part of the DBMR method. The BRE module measures the correlation between target and auxiliary behaviors. By setting behavioral correlation scores c in Eq. (13) to identical values, this variant gives equal importance to recommendation tasks across different behaviors during multi-task learning.

Fig. 5 presents the performance comparison between the DBMR method and its two variants on the Tmall and Taobao datasets. The experimental results demonstrate that the removal of either attention mechanism leads to a decline in DBMR's performance on both H@K and N@K evaluation metrics, validating the critical role of attention mechanisms in the model. Specifically, the maximum performance reductions observed on the Tmall dataset reach 2.07% and 3.32%, while on the Taobao dataset they reach 6.44% and 11.21%, respectively. These experimental results confirm that using uniform weights fails to achieve optimal performance, whereas explicitly learning the distinct contributions of different auxiliary behaviors to target behavior prediction through differentiated weighting strategies significantly enhances the model's effectiveness.

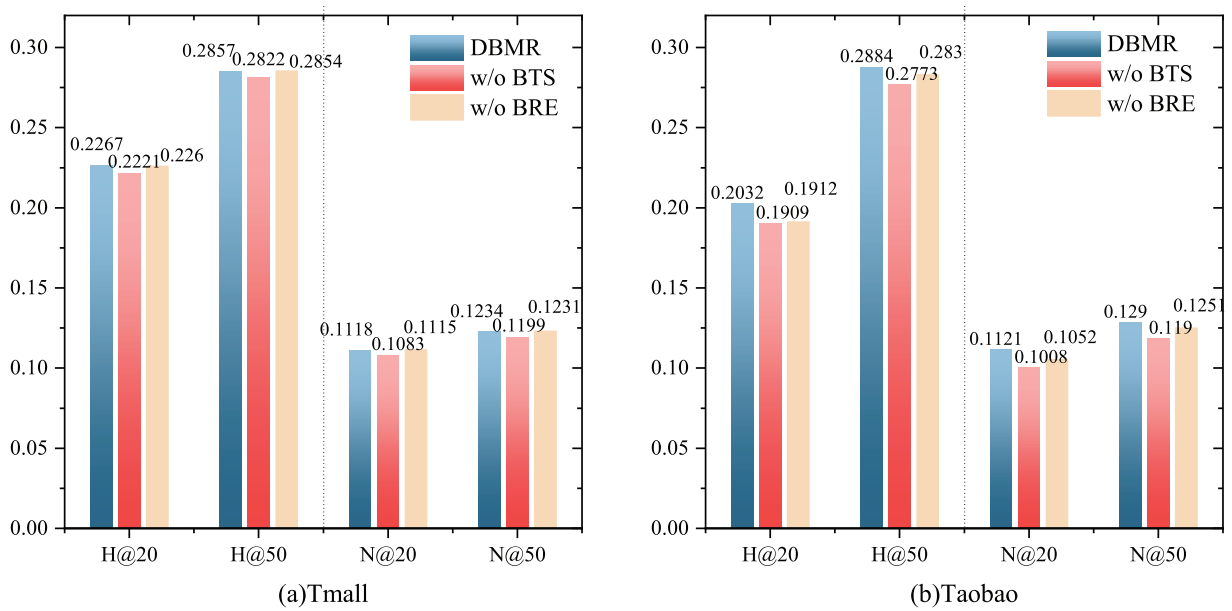

Figure 5: The ablation experiment of DBMR attention mechanism.

5.3.2 Item Semantic Alignment Constraint Module

To validate the effectiveness of introducing the Semantic Alignment Constraint (SAC), a model variant designated as w/o SAC was generated by eliminating this constraint. This section compares the DBMR method with the w/o SAC variant.

Fig. 6 presents a comparative analysis of the recommendation performance between DBMR and the w/o SAC variant across both datasets. The experimental results indicate that the w/o SAC variant exhibits varying degrees of performance degradation in predictive capability on both datasets. These findings demonstrate the critical role of semantic alignment between attribute features and behavioral paths in enhancing recommendation accuracy. By integrating static item attributes with dynamic behavioral patterns within the representation space, the model learns more discriminative feature representations, effectively mitigates data sparsity, and enhances the ability to capture users' genuine intents, thereby significantly improving the precision and robustness of recommendations.

5.4 Parameter Analysis

To analyze the impact of LightGCN network depth on model performance, this section adopts different numbers of GCN propagation layers for page-view and favourite behaviors, with values selected from 1, 2, 3, 4. The corresponding layer counts were independently adjusted for page-view and favourite behaviors, enabling DBMR to achieve optimal performance on the NDCG@10 evaluation metric across both datasets. The experimental results are shown in Fig. 7.

From the experimental results, it can be observed that on the Tmall and Taobao datasets, the DBMR model demonstrates optimal performance when the GCN propagation layer configurations for the (*page-view* → *favourite*) behavior are set to (2, 4) and (1, 4), respectively. Further analysis reveals that as the number of GCN layers for page-view behavior increases, model performance declines. This may be attributed to the amplification of interaction noise inherent in page-view behaviors (such as accidental clicks or casual page views) during deep propagation, thereby interfering with the embedding learning of target

behaviors. Conversely, increasing the number of GCN layers on the favourite behavior graph consistently yields performance improvements, indicating that multi-layer propagation effectively aggregates multi-hop neighbor information from the favourite behavior graph. This enables the learning of more discriminative embedding representations, thereby providing a more reliable foundation for final preference prediction.

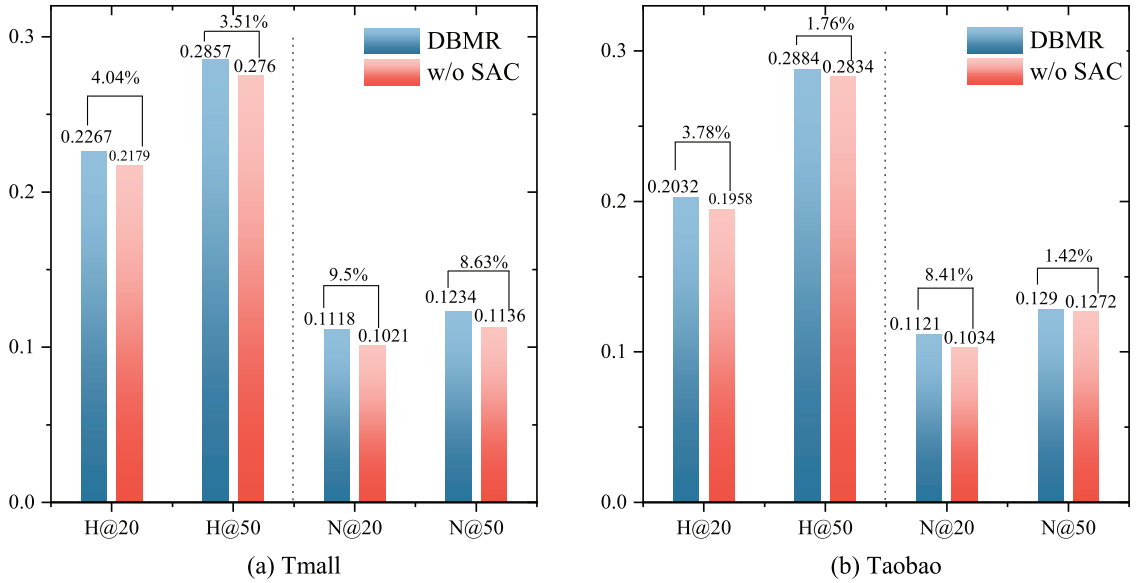

Figure 6: The ablation experiment of DBMR item semantic alignment constraint.

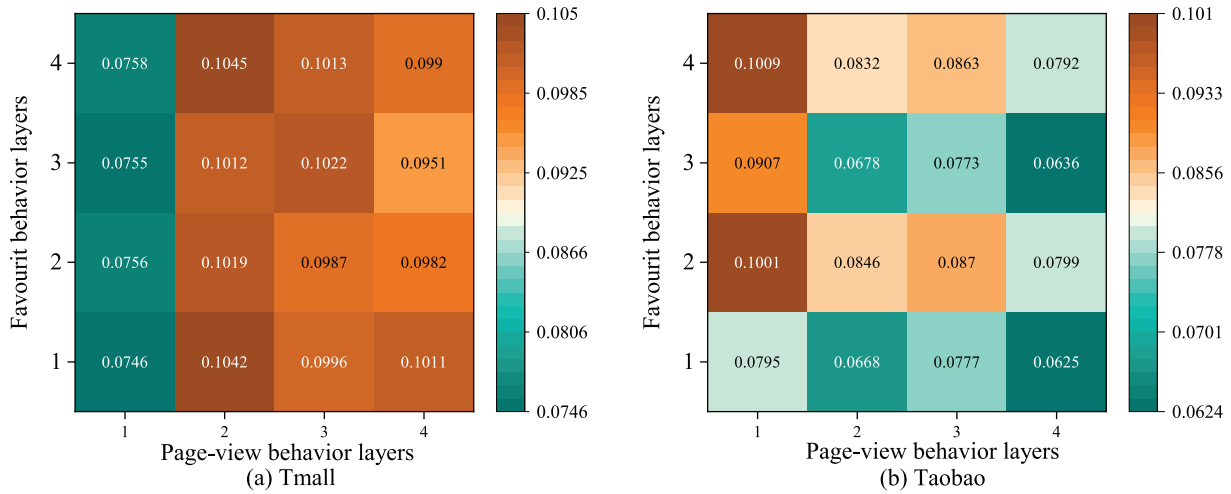

Figure 7: Impact of GNN Layers for different behaviors.

6 Future Research Directions

Currently, the DBMR model employs a static approach to construct behavioral pathway maps, which makes it difficult to effectively capture the dynamic changes in user interests and fine-grained temporal behavior patterns. To enhance the model's dynamic adaptability [38], future research will focus on advancing from the following two aspects: First, we will explore methods for constructing and modeling the evolution of temporal behavior graphs. Specifically, user-item interactions can be segmented based on time windows

to construct dynamic graph sequences, and temporal graph neural networks or similar techniques can be utilized to model the changes in graph structure. This approach can directly capture trends in interest drift and periodic behavioral patterns. Second, we will introduce a sequence-based attention mechanism into the cross-behavior message aggregation module. This mechanism takes the user's historical behavior sequence as input, captures dependencies between behaviors through temporal encoding, and dynamically generates an attention weight for each historical behavior b_{k-1} that influences the current behavior b_k, thereby explicitly modeling temporal dependencies.

Additionally, the current DBMR method fails to capture the influence of temporal proximity between behaviors on behavioral transition strength. For instance, a "page view →purchase" behavior chain occurring within the same session should inherently exhibit a much stronger signal than a similar chain separated by several weeks. Future research will enhance temporal awareness by introducing a time decay function when calculating transition strength. This will assign a decay weight, correlated with the time interval, to each behavioral transition edge (b_{k-1}, b_k). This adjustment will enable the model to assign greater importance to recently occurring behavioral conversions, thereby more accurately reflecting the user's current interests.

7 Conclusion

We propose a Directed Behavior path graph-based Multi-behavior Recommendation method (DBMR). The method constructs a directed behavior path graph for users and items based on diversified behavior chains, and builds user-item interaction graphs for each behavior type. On this basis, a LightGCN model with a residual design is used to learn the feature embeddings of users and items. Then, a graph attention message aggregator is used to perform weighted feature aggregation. Finally, the recommendation score is calculated based on the user preferences from terminating behaviors and the item feature representations. In terms of model optimization, this paper introduces multi-task learning and an item-semantic alignment constraint to jointly optimize across multiple tasks. Experimental results on two real-world datasets validate the superiority of the DBMR model. In this paper, we not only explore the potential relationships between user behavior paths and item attributes but also demonstrate the diversity of behavior paths.

Acknowledgement: We are thankful for all contributors.

Funding Statement: This research was supported by the Key Research and Development Program of China (Grant No. 2022YFB3102904).

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Qian Hu; methodology, Qian Hu and Lei Tan; software, Qian Hu; validation, Qian Hu; formal analysis, Qingjun Yuan; investigation, Qian Hu and Yan Li; resources, Zong Zuo; data curation, Qian Hu; writing—original draft preparation, Qian Hu; writing—review and editing, Qian Hu; visualization, Lei Tan; project administration, Qingjun Yuan; funding acquisition, Lei Tan. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data openly available in a public repository. 1. "The data that support the findings of this study are openly available in Tmall at https://www.tmall.com." 2."The data that support the findings of this study are openly available in Taobao at https://tianchi.aliyun.com/dataset/dataDetail?dataId=649".

Ethics Approval: Not applicable.

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

References

1. He X, Liao L, Zhang H, Nie L, Hu X, Chua TS. Neural collaborative filtering. In: 2017 Proceedings of the 26th International Conference on World Wide Web; 2017 Apr 3–7; Perth, Australia. Geneva, Switzerland: International World Wide Web Conferences Steering Committee; 2017. p. 173–82. doi:10.1145/3038912.3052569.
2. Lin F, Zhao Z, Zhu X, Zhang D, Shen S, Li X, et al. When box meets graph neural network in tag-aware recommendation. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2024 Aug 25–29; Barcelona, Spain. New York, NY, USA: ACM; 2024. p. 1770–80. doi:10.1145/3637528.3671973.
3. 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; 2020 Jul 25–30; Xi'an, China. New York, NY, USA: ACM; 2020. p. 639–48. doi:10.1145/3397271.3401063.
4. Chen C, Zhang M, Zhang Y, Ma W, Liu Y, Ma S. Efficient heterogeneous collaborative filtering without negative sampling for recommendation. In: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence; 2020 Feb 7–12; New York, USA. p. 19–26. doi:10.1609/aaai.v34i01.5329.
5. Wu B, Cheng Y, Yuan H, Ma Q. When multi-behavior meets multi-interest: multi-behavior sequential recommendation with multi-interest self-supervised learning. In: Proceedings of the 2024 IEEE 40th International Conference on Data Engineering (ICDE); 2024 May 13–17; Utrecht, The Netherlands. Piscataway, NJ, USA: IEEE; 2024. p. 845–58. doi:10.1109/ICDE60146.2024.00070.
6. Xia L, Huang C, Xu Y, Dai P, Lu M, Bo L. Multi-behavior enhanced recommendation with cross-interaction collaborative relation modeling. In: Proceedings of the 2021 IEEE 37th International Conference on Data Engineering (ICDE); 2021 Apr 19–22; Chania, Greece. Piscataway, NJ, USA: IEEE; 2021. p. 1931–6. doi:10.1109/ICDE51399.2021.00179.
7. 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; 2020 Jul 25–30; Xi'an, China. New York, NY, USA: ACM; 2020. p. 2397–406. doi:10.1145/3397271.3401445.
8. Xue H, Yang L, Rajan V, Jiang W, Wei Y, Lin Y, et al. Multiplex bipartite network embedding using dual hypergraph convolutional networks. In: Proceedings of the Web Conference 2021; 2021 Apr 19–23; Ljubljana, Slovenia. New York, NY, USA: ACM; 2021. p. 1649–60. doi:10.1145/3442381.3449954.
9. Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer. 2009;42(8):30–7. doi:10.1109/MC.2009.263.
10. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L. BPR: bayesian personalized ranking from implicit feedback. In: 2009 Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009); 2009 Jun 18–21; Montreal, QC, Canada. Corvallis, OR, USA: AUAI Press; 2009. p. 452–61. doi:10.48550/arXiv.1205.2618.
11. Ko G, Jung J. Learning disentangled representations in signed directed graphs without social assumptions. Inf Sci. 2024;665(7):120373. doi:10.1016/j.ins.2024.120373.
12. Seo C, Jeong KJ, Lim S, Shin WY. SiReN: sign-aware recommendation using graph neural networks. IEEE Trans Neural Netw Learn Syst. 2024;35(4):4729–43. doi:10.1109/TNNLS.2022.3175772.
13. Lee JW, Jung J. Time-aware random walk diffusion to improve dynamic graph learning. In: Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI-23); 2023 Feb 7–14; Washington, DC, USA. Palo Alto, CA, USA: AAAI Press; 2023. p. 8473–81.
14. Park H, Jung J, Kang U. A comparative study of matrix factorization and random walk with restart in recommender systems. arXiv:1708.09088. 2017. doi:10.48550/arXiv.1708.09088.
15. Xia L, Huang C, Xu Y, Dai P, Zhang X, Yang H, et al. Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI-21); 2021 Feb 2–9; Virtual Event. Palo Alto, CA, USA: AAAI Press; 2021. p. 4486–93. doi:10.48550/arXiv.2110.04000.

16. 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; 2020 Jul 25–30; Xi'an, China. New York, NY, USA: ACM; 2020. p. 659–68. doi:10.1145/3397271.3401072.
17. Tang L, Long B, Chen BC, Agarwal D. An empirical study on recommendation with multiple types of feedback. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2016 Aug 13–17; San Francisco, CA, USA. New York, NY, USA: ACM; 2016. p. 283–92. doi:10.1145/2939672.2939690.
18. Yan M, Cheng Z, Sun J, Sun F, Peng Y. MB-HGCN: a hierarchical graph convolutional network for multi-behavior recommendation. arXiv:2306.10679. 2023. doi:10.48550/arXiv.2306.10679.
19. Meng C, Zhang H, Guo W, Guo H, Liu H, Zhang Y, et al. Hierarchical projection enhanced multi-behavior recommendation. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2023 Aug 6–10; Long Beach, CA, USA. New York, NY, USA: ACM; 2023. p. 4649–60. doi:10.1145/3580305.3599838.
20. Liu S, Liang Y, Gitter A. Loss-balanced task weighting to reduce negative transfer in multi-task learning. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI-19); 2019 Jan 27–Feb 1; Honolulu, HI, USA. Palo Alto, CA, USA: AAAI Press; 2019. p. 9977–8. doi:10.1609/aaai.v33i01.33019977.
21. 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 (CIKM '23); 2023 Oct 21–25; Birmingham, UK. New York, NY, USA: ACM; 2023. p. 1797–806. doi:10.1145/3583780.3615004.
22. Wan M, McAuley J. Item recommendation on monotonic behavior chains. In: Proceedings of the 12th ACM Conference on Recommender Systems; 2018 Oct 2–7; Vancouver, BC, Canada. New York, NY, USA: ACM; 2018. p. 86–94. doi:10.1145/3240323.3240369.
23. Gao C, He X, Gan D, Chen X, Feng F, Li Y, et al. Learning to recommend with multiple cascading behaviors. IEEE Trans Knowl Data Eng. 2019;33(6):2588–601. doi:10.1109/TKDE.2019.2958808.
24. Yan M, Cheng Z, Gao C, Sun J, Liu F, Sun F, et al. Cascading residual graph convolutional network for multi-behavior recommendation. ACM Trans Inf Syst. 2023;42(1):1–26. doi:10.1145/3587693.
25. Cheng Z, Han S, Liu F, Zhu L, Gao Z, Peng Y. Multi-behavior recommendation with cascading graph convolution networks. In: 2023 Proceedings of the ACM Web Conference 2023 (WWW '23); 2023 Apr 30–May 4; Austin, TX, USA. New York, NY, USA: ACM; 2023. p. 1181–9. doi:10.1145/3543507.3583439.
26. Zhu X, Lin F, Zhao Z, Xu T, Zhao X, Yin Z, et al. Multi-behavior recommendation with personalized directed acyclic behavior graphs. ACM Trans Inf Syst. 2024;43(1):1–30. doi:10.1145/3696417.
27. Yan M, Liu F, Sun J, Sun F, Cheng Z, Han Y. Behavior-contextualized item preference modeling for multi-behavior recommendation. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2024 Jul 14–18; Washington, DC, USA. New York, NY, USA: ACM; 2024. p. 946–55. doi:10.1145/3626772.3657696.
28. Liu H, Huang X, Zhou W, Luo F, Wen J, Zhang H. Representation-enhanced cascading multi-level interest learning for multi-behavior recommendation. ACM Trans Inf Syst. 2025. doi:10.1145/3786341.
29. Lu D, Wu S, Zhang H, Xu G, Han Q. Causal cascading convolution networks for multi-behavior sequential recommendation. Inf Sci. 2025;720(4):122484. doi:10.1016/j.ins.2025.122484.
30. Yang J, Li X, Li B, Tian L, Xu B, Chen Y. Cascading graph contrastive learning for multi-behavior recommendation. Neurocomputing. 2024;610(9):128618. doi:10.1016/j.neucom.2024.128618.
31. Wei W, Huang C, Xia L, Xu Y, Zhao J, Yin D. Contrastive meta learning with behavior multiplicity for recommendation. In: Proceedings of the 15th ACM International Conference on Web Search and Data Mining; 2022 Feb 21–25; Virtual Event. New York, NY, USA: ACM; 2022. p. 1120–8. doi:10.1145/3488560.3498527.
32. Li P, Li R, Da Q, Zeng AX, Zhang L. Improving multi-scenario learning to rank in e-commerce by exploiting task relationships in the label space. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management; 2020 Oct 19–23; Virtual Event, Ireland. New York, NY, USA: ACM; 2020. p. 2605–12. doi:10.1145/3340531.3412713.

33. Su L, Pan J, Wang X, Xiao X, Quan S, Chen X, et al. STEM: unleashing the power of embeddings for multi-task recommendation. In: Proceedings of the 38th AAAI Conference on Artificial Intelligence; 2024 Feb 20–27; Vancouver, BC, Canada. Palo Alto, CA, USA: AAAI Press; 2024. p. 9002–10. doi:10.1609/aaai.v38i8.28749.
34. Li J, Wang M, Li J, Fu J, Shen X, Shang J, et al. Text is all you need: learning language representations for sequential recommendation. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2023 Aug 6–10; Long Beach, CA, USA. New York, NY, USA: ACM; 2023. p. 1258–67. doi:10.1145/3580305.3599519.
35. Gao C, He X, Gan D, Chen X, Feng F, Li Y, et al. Neural multi-task recommendation from multi-behavior data. In: Proceedings of the 2019 IEEE 35th International Conference on Data Engineering; 2019 Apr 8–11; Paris, France. Piscataway, NJ, USA: IEEE; 2019. p. 1554–7. doi:10.1109/ICDE.2019.00140.
36. 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; 2021 Jul 11–15; Montreal, QC, Canada. New York, NY, USA: ACM; 2021. p. 757–66. doi:10.1145/3404835.3462972.
37. Gu S, Wang X, Shi C, Xiao D. Self-supervised graph neural networks for multi-behavior recommendation. In: Proceedings of the 31st International Joint Conference on Artificial Intelligence; 2022 Jul 23–29; Vienna, Austria. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 2022. p. 2052–8. doi:10.24963/ijcai.2022/285.
38. Wan S, Yang S, Fu Z. Focus on user micro multi-behavioral states: time-sensitive user behavior conversion prediction and multi-view reinforcement learning based recommendation approach. Inf Process Manag. 2025;62(2):103967. doi:10.1016/j.ipm.2024.103967.