Neurocomputing 651 (2025) 130952

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

CMC-GCN: Consistent multi-granularity cascading graph convolution network for multi-behavior recommendation

Yabo Yin[a], Xiaofei Zhu[a,*], Kunyang Huang[b], Wenshan Wang[c], Yihao Zhang[d], Pengfei Wang[e], Yixing Fan[c], Jiafeng Guo[c]

[a] College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China
[b] College of Science, Mathematics and Technology, Wenzhou-Kean University, Wenzhou 325060, China
[c] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[d] School of Artificial Intelligence, Chongqing University of Technology, Chongqing, 400054, China
[e] School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

HIGHLIGHTS

- Introduce a novel multi-granularity cascading graph convolutional network.
- Propose a behavior consistency-guided alignment strategy to maintain both intra- and inter-behavior consistency.
- Conduct extensive experiments on four real-world datasets to examine model's effectiveness.

ARTICLE INFO

Communicated by C. Gao

Keywords:
Multi-behavior recommendation
Self-supervised learning
Graph convolutional network

ABSTRACT

Multi-behavior recommendation (MBR) has garnered growing attention recently due to its ability to mitigate the sparsity issue by inferring user preferences from various auxiliary behaviors to improve predictions for the target behavior. Although existing research on MBR has yielded impressive results, it still faces two major limitations. First, previous methods mainly focus on modeling fine-grained interaction information between users and items under each behavior, which may suffer from sparsity issue. Second, existing models usually concentrate on exploiting dependencies between two consecutive behaviors, leaving intra- and inter-behavior consistency largely unexplored. To this end, we propose a novel approach named Consistent Multi-Granularity Cascading Graph Convolution Network for Multi-Behavior Recommendation (CMC-GCN). To be specific, we first explore both fine- and coarse-grained correlations among users or items of each behavior by simultaneously modeling the behavior-specific interaction graph and its corresponding hypergraph in a cascaded manner. Then, we propose a behavior consistency-guided alignment strategy that preserves consistent representations between the interaction graph and its associated hypergraph for each behavior, while also maintaining representation consistency across different behaviors. Extensive experiments and analyses on four real-world benchmark datasets demonstrate that our proposed approach is consistently superior to previous state-of-the-art methods due to its capability to effectively attenuate the sparsity issue as well as keep both intra- and inter-behavior consistencies. The code is available at https://github.com/marqu22/CMCGCN.git.

1. Introduction

Multi-behavior recommendation (MBR) has emerged as an important research problem in recent years and has garnered significant attention across various communities. Compared to traditional recommendation methods [1–3], which depend exclusively on one behavior pattern, MBR makes recommendations by leveraging diverse and detailed interaction information between users and items to alleviate the data sparsity issue. Besides the target behavior (e.g., buy), multiple supporting behaviors

* Corresponding author.
Email addresses: yinyabo@stu.cqut.edu.cn (Y. Yin), zxf@cqut.edu.cn (X. Zhu), huangku@kean.edu (K. Huang), wangwenshan@ict.ac.cn (W. Wang), yhzhang@cqut.edu.cn (Y. Zhang), wangpengfei@bupt.edu.cn (P. Wang), fanyixing@ict.ac.cn (Y. Fan), guojiafeng@ict.ac.cn (J. Guo).

https://doi.org/10.1016/j.neucom.2025.130952
Received 29 April 2025; Received in revised form 28 June 2025; Accepted 6 July 2025
Available online 8 July 2025

(e.g., view, click and cart) contribute to improving recommendation effectiveness.

Early studies [4,5] can be regarded as straightforward extensions of single-behavior recommendation models, aiming to better represent users and items in the target behavior by utilizing shared embeddings across various behaviors. Meanwhile, some findings [6,7] reveal that certain sampling strategies can boost model recommendation performance. Later, many research efforts have been devoted to leveraging Deep Neural Networks (DNNs) for the MBR task due to their strong capability to capture the complex relationships between users and items, resulting in encouraging performance. These include methods that employ hierarchical attention mechanism to model inner parts within a behavior as well as inter-view relations between different behaviors [8], or transfer the prediction of low-level behaviors to high-level behaviors [9].

Recently, a number of graph convolutional networks (GCNs) based methods have been proposed to explicitly capture the high-order dependencies between diverse types of user-item interaction data by utilizing the graph-based message passing mechanism. For example, MBGCN [10] leverages multi-behavior data to construct a unified graph, and applies both user-item propagation and item–item propagation to learn behavior strength and behavior semantics, respectively. GNMR [11] builds a multi-behavior interaction graph and performs embedding propagation to capture complex relationships among different types of user behaviors. S-MBRec [12] introduces a supervised signal to consider the differences among various behaviors and develops a target behavior-centered contrastive learning approach to capture commonality between auxiliary and target behaviors. MB-HGCN [13] learns behavior-specific and global embeddings based on each individual behavior graph and a unified homogeneous graph. The multi-task learning strategy is leveraged for joint optimization. BCIPM [14] considers the variation of user preferences across different behaviors and distills item-aware preferences from user-item interactions. Some studies [15–18] propose exploiting the cascading relationship in diverse behavioral patterns to enhance the performance of recommendation, where useful information learned from preceding behaviors is utilized to refine the learned embeddings of successive behaviors.

Despite the encouraging progress obtained by conventional methods, there are still two key issues limiting their efficacy. Firstly, since user interaction data exhibits sparsity and the bipartite graph representing user-item relationships has inherent structural constraints, employing multi-layer graph message propagation can easily lead to the over-smoothing problem, resulting in model performance degradation. Existing research works mainly rely on modeling fine-grained interactions between users and items under specific behaviors, while they overlook exploring the coarse-grained information existing in the interaction graph, e.g., certain users may have similar interest preferences, or specific items may belong to the same category. Since the coarse-grained information can effectively complement the fine-grained interaction information, leveraging both simultaneously is valuable for capturing more accurate user preferences. Secondly, previous cascading based models [16,18] emphasize exploiting dependencies in the sequence of behaviors for embedding learning, where the embeddings derived from preceding behaviors serve as input for subsequent behavior representation learning. Although they can effectively capture the dependency among two consecutive behaviors, the rich global consistency among different behaviors as well as the intra-behavior dependency signals is largely ignored.

In response to the previously identified limitations, we develop an innovative approach designated as **C**onsistent **M**ulti-Granularity **C**ascading **G**raph **C**onvolution **N**etwork for Multi-Behavior Recommendation (CMC-GCN). Specifically, we first propose a multi-granularity cascading graph convolution network by introducing a user-item interaction graph and a hypergraph for each behavior. The behavior-specific hypergraph is developed to capture coarse-grained correlations among users or items, which can alleviate the sparsity issue within the user-item interaction graph. Then, we develop a behavior consistency-guided contrastive learning module, which consists of an intra-behavior consistency contrastive learning submodule and a cross-behavior consistency contrastive learning submodule. The former aims to align user and item embeddings inferred from an interaction graph to their counterparts learned from the corresponding hypergraph. The latter attempts to align embeddings of users and items from each behavior towards the common embeddings derived from a global graph with user-item interactions from all behaviors. Our method is rigorously evaluated using four real-world datasets: Beibei, Taobao, Tmall and QKV. CMC-GCN shows superior results compared to all baseline approaches, demonstrating HR@10 improvements of 19.20 %, 37.45 %, 13.43 % and 13.57 % relative to the best baseline on these four datasets.

In conclusion, the key contributions include:

- We introduce a novel multi-granularity cascading graph convolution network, which explores both fine- and coarse-grained correlations among users or items by simultaneously modeling the user-item interaction graph and its corresponding hypergraph for each behavior.
- We propose a behavior consistency-guided alignment strategy to maintain both intra- and inter-behavior consistency based on contrastive learning.
- We perform detailed empirical studies on four real-world datasets to verify CMC-GCN's capabilities. The obtained results exhibit substantial progress relative to advanced baseline models.

2. Related work

Multi-behavior recommendation seeks to utilize valuable signals from auxiliary behaviors to mitigate the sparsity issue in the target behavior. Early methods [4,5,19] mainly rely on extending the single-behavior recommendation technique such as matrix factorization by introducing shared embeddings across different behaviors. Some research [6,7,20] applies sampling strategies to incorporate information from multiple auxiliary behaviors, which are tailored to inherent characteristics in multi-behavior scenarios.

With the remarkable success of deep neural networks (DNNs) in recommendation systems, numerous approaches leveraging DNNs have been developed to capture user preferences through identifying sophisticated associations among users and items [21]. For instance, DIPN [8] develops a hierarchical modeling approach composed of dual attention layers: the lower layer captures intra-behavior part-level interactions, while the upper layer learns the inter-view relations across distinct behaviors. EHCF [9] attempts to transfer the prediction of low-level behaviors to high-level behaviors to capture the complicated relationships among different behaviors. ARGO [22] captures the characteristics of each user by developing a neural collaborative filtering model and models the ordinal relation among multiple behaviors by correlating each behavior's prediction.

Lately, graph convolutional networks (GCNs) have received significant attention in multi-behavior recommendation research [23–26]. These models typically represent user-item interactions through graph-based structures and employ message propagation mechanisms to learn node representations. For instance, MBGCN [10] integrates multi-behavior data through a unified graph structure and develops a multi-behavior graph convolutional network. It utilizes dual propagation mechanisms between user-item and item–item to learn behavioral intensity and semantic patterns, respectively. GNMR [11] captures latent relationships between behaviors using a relational dependency encoder, and represents graph-structured interactions through propagated embeddings across a heterogeneous behavior graph. Inspired by the success of contrastive learning in representation learning [27–29], S-MBRec [12] incorporates both a supervised task for capturing embedding discrepancies and a target behavior-centered contrastive learning mechanism to model the commonalities between auxiliary and target behaviors. MB-HGCN [13] employs a hierarchical graph convolutional network which consists of global embedding learning and behavior-specific embedding learning. Since direct information transfer between auxiliary and target behaviors may introduce noise given the divergence in user preferences among distinct behavior types. KMCLR [30] introduces a dual contrastive learning framework, complemented by a knowledge-enhanced module that optimizes item representation learning. KDMBR [31] utilizes a behavior-aware attention mechanism to weigh auxiliary behaviors dynamically and integrates knowledge graphs to enrich item representations. BCIPM [14] proposes deriving item-aware preferences from user-item interactions over various behavior types, while only considering preferences associated with the target behavior in recommendation. DeMBR [32] employs a pruning-based denoising module to eliminate data-level noise, coupled with a semantic guidance denoising module that suppresses representation-level noise while preserving true user preferences.

As there are usually certain orders between different behaviors, some research work concentrates on capturing such behavior dependencies to learn better user preferences. NMTR [15] learns user preferences by assuming that there is a shared embedding across different behaviors, and explores the ordinal relationship information by correlating the model prediction in a cascaded manner. CRGCN [18] proposes a cascading residual graph convolutional network to model user preferences by exploiting the connections between consecutive behaviors, and adopts the multi-task learning to conduct joint optimization by comprehensively exploiting supervision signals from different behaviors. MB-CGCN [16] explicitly models the behavior dependencies for learning user preferences, and their aggregation facilitates the final behavior prediction. PKEF [17] extends the cascade paradigm by incorporating parallel knowledge to learn better representations for different behaviors, simultaneously acquiring hierarchical correlation information to mitigate challenges posed by imbalanced behavior distribution. COPF [33] introduces a combinatorial optimization graph convolutional network to correct behavior pattern modeling errors from limited views, and employs a distribution-fitting expert network to reduce negative transfer from uncoordinated tasks.

Unlike the aforementioned studies, our approach introduces several key innovations to comprehensively model user preferences. First, existing methods primarily rely on pairwise user-item interactions within a single graph structure to learn fine-grained user preferences, while often overlooking the coarse-grained correlations inherent within each behavior. To address this, we incorporate a behavior-specific hypergraph to explicitly capture these coarse-grained correlations among users or items for each behavior type. This enables our model to simultaneously leverage both fine-grained interaction signals and coarse-grained relational information for a more comprehensive representation. Second, differing from methods concentrating solely on target-auxiliary behavior consistency (e.g., S-MBRec [12]) or employing sequential dependency modeling via a cascading structure (e.g., CRGCN [18], PKEF [17]), we develop a novel behavior consistency-guided contrastive learning mechanism. This mechanism is designed to simultaneously maintain both intra-behavior consistency (within each individual behavior) and global inter-behavior consistency (across all behaviors), leading to more robust and informative preference representations.

3. Problem formulation

In this work, we concentrate on constructing an innovative recommendation framework that leverages auxiliary behaviors to improve the acquisition of more accurate user preferences for the target behavior. We describe the basic notations frequently used in this paper. Let $\mathcal{U} = \{u_1, u_2, \dots, u_M\}$ and $\mathcal{I} = \{i_1, i_2, \dots, i_N\}$ denote the set of users and items, where M and N denote the number of users and items, respectively. We use k ($1 \le k \le K$) to denote the k-th behavior, where K denotes the number of users' behaviors. $\mathcal{G}_k$ is the user-item interaction graph for the k-th behavior, and $\mathcal{R}^k$ is the interaction matrix of the corresponding behavior. For $r^k_{ui} \in \mathcal{R}^k$, $r^k_{ui} = 1$ indicates that there is an interaction between user u and item i under the k-th behavior, otherwise $r^k_{ui} = 0$. In addition, we define a global graph $\mathcal{G}_g = \cup_{k=1}^{K} \mathcal{G}_k$ and its corresponding interaction matrix is $\mathcal{R}^g$. For $r^g_{ui} \in \mathcal{R}^g$, $r^g_{ui} = 1$ means that user u and item i have interacted at least once in these K behaviors. The MBR task is then formulated as follows:

Input: A user set $\mathcal{U}$, an item set $\mathcal{I}$, a list of behavior interaction graphs $[\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_K]$, and a global graph $\mathcal{G}_g$.

Output: A predicted score that estimates the interaction probability between user u and item i for the designated K-th behavior.

4. Proposed method

We propose a Consistent Multi-Granularity Cascading Graph Convolution Network (CMC-GCN) for multi-behavior recommendation, with four core modules including: (1) A *Global Graph Learning* module, which is dedicated to acquiring insights into users' general preferences across distinct behaviors. (2) A *Multi-Granularity Cascading Graph Convolution Network*, which follows the cascading paradigm and incorporates behavior hypergraphs to strengthen behavioral-specific representation quality. (3) A *Behavior Consistency-Guided Contrastive Learning* module encompassing both intra- and inter-behavior intrinsic consistency. (4) A *Prediction* module that employs multi-task learning to conduct joint optimization (Fig. 1).

4.1. Global graph learning

We introduce the global graph $\mathcal{G}_g$ to serve as an indicator of whether user u and item i have interacted at least once across all behaviors, i.e., $\mathcal{G}_g$ contains all user interactions across multiple behaviors. We utilize Xavier [34] to initialize user u and item i as d-dimensional embeddings, denoted as $e^0_u \in \mathbb{R}^d$, $e^0_i \in \mathbb{R}^d$, respectively. Then, we employ LightGCN [1] to refine the representations of both users and items as follows:

$$e^{(l)}_u = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} e^{(l-1)}_i, \tag{1}$$

$$e^{(l)}_i = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} e^{(l-1)}_u, \tag{2}$$

where $\mathcal{N}_u$ represents all items with which user u has engaged, while $\mathcal{N}_i$ corresponds to all users who have interacted with item i. $e^{(l)}_u$ and $e^{(l)}_i$ denote the embeddings of user u and item i after the graph convolution of the l-layer, respectively. $e^{(0)}_u$ and $e^{(0)}_i$ are initialized with e^0_u and e^0_i, respectively.

We obtain the global embedding representations of user u and item i by aggregating the outputs of all layers:

$$e^g_u = \sum_{l=0}^{L} e^{(l)}_u, \tag{3}$$

$$e^g_i = \sum_{l=0}^{L} e^{(l)}_i, \tag{4}$$

where L is the total number of graph convolutional layers.

4.2. Multi-granularity cascading graph convolution network

Our framework introduces a multi-granularity cascading graph convolutional network to learn fine- and coarse-grained user and item representations from the user-item interaction graph and its associated hypergraph for each behavior in a cascaded manner.

4.2.1. Behavior-specific interaction graph learning

For the k-th behavior, we employ LightGCN on its corresponding user-item interaction graph $\mathcal{G}_k$ to produce node representations. Specifically, we update node representations by aggregating information from their neighbors as follows:

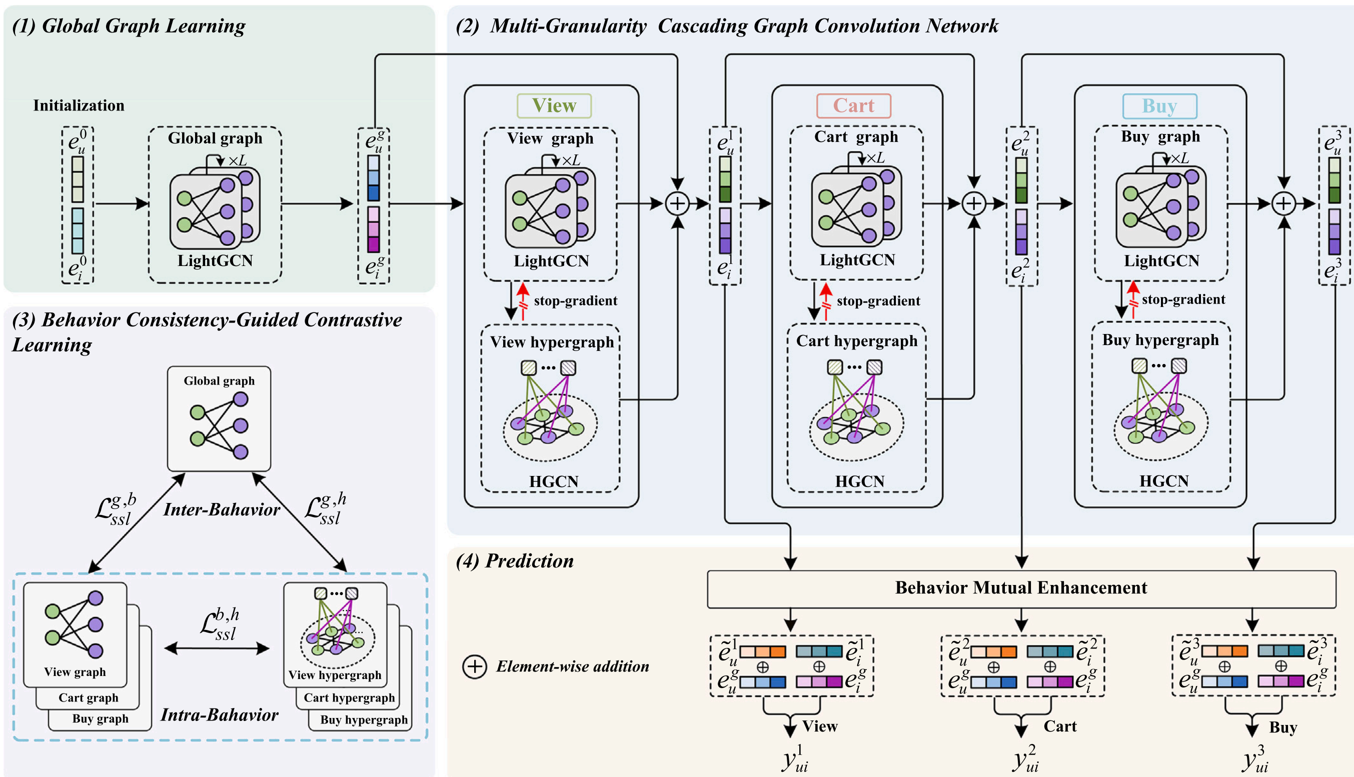


Fig. 1. The overall architecture of CMC-GCN. We utilize three behaviors (i.e., view, cart and buy) as an example, wherein buy is the target behavior.

$$\mathbf{e}_u^{k,(l)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} \mathbf{e}_i^{k,(l-1)}, \tag{5}$$

$$\mathbf{e}_i^{k,(l)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^{k,(l-1)}, \tag{6}$$

where $\mathbf{e}_u^{k,(l)}$ and $\mathbf{e}_i^{k,(l)}$ respectively denote the updated representations of user u and item i after l layer propagation on the k-th interaction graph. Note that for the k-th behavior-specific interaction graph, we initialize the embeddings for both users and items using those obtained from their prior behavior:

$$e_u^{k,(0)} = e_u^{k-1}, \tag{7}$$

$$e_i^{k,(0)} = e_i^{k-1}, \tag{8}$$

where e_u^{k-1} and e_i^{k-1} indicate the user and item vector representations generated by the preceding behavior-specific interaction graph. Meanwhile, we initialize the embeddings for the first behavior using the learned user and item embeddings that originate from global graph learning. We then calculate the embedding vectors for user u and item i specific to the k-th behavior:

$$e_u^{b,k} = \sum_{l=0}^{L} e_u^{k,(l)}, \tag{9}$$

$$e_i^{b,k} = \sum_{l=0}^{L} e_i^{k,(l)}. \tag{10}$$

4.2.2. Behavior-specific hypergraph learning

In traditional bipartite graphs, edges are confined to pairwise interactions. This constraint impedes the learning of high-order correlations, consequently masking meaningful patterns like shared preferences among user groups or category relationships among items. To capture the complex high-order correlations among users or items, we augment our approach by introducing additional learnable behavior-specific hypergraphs. Unlike conventional graph structures, a hypergraph enables each edge to associate with multiple nodes [35], thus modeling latent relationships among groups of users or items beyond hop distance limit.

For hypergraph construction, we employ a parameterized hypergraph structure learning method, which dynamically learns hyperedge representations during training to adaptively capture the hypergraph dependency structure among users and items. Specifically, we first define a set of hyperedge representations $\mathcal{H}_U^k \in \mathbb{R}^{M \times S}$ for all M users on the k-th behavior, in which S indicates the total hyperedges. Similarly, the hyperedge representations for all N items are defined as $\mathcal{H}_I^k \in \mathbb{R}^{N \times S}$. Inspired by the prior work [36], we address computational efficiency and mitigate overfitting concerns within the hypergraph learning module. Specifically, we employ parameterized hyperedges, which facilitate hyperedge parameter optimization in a low-rank manner. This operation is formulated as:

$$\mathcal{H}_U^k = E_U^{b,k} \cdot W_U^{h,k}, \tag{11}$$

$$\mathcal{H}_I^k = E_I^{b,k} \cdot W_I^{h,k}, \tag{12}$$

where $W_U^{h,k} \in \mathbb{R}^{d \times S}$ and $W_I^{h,k} \in \mathbb{R}^{d \times S}$ are the learnable parameters, $E_U^{b,k} = [e_{u_1}^{b,k}, \ldots, e_{u_M}^{b,k}] \in \mathbb{R}^{M \times d}$ and $E_I^{b,k} = [e_{i_1}^{b,k}, \ldots, e_{i_N}^{b,k}] \in \mathbb{R}^{N \times d}$ represent the embeddings of all users and items corresponding to the k-th behavior-specific interaction graph, respectively. For hypergraph learning in the k-th behavior, we utilize a simplified hypergraph convolution network (HGCN) [37] for the user-side and item-side hypergraphs of each behavior:

$$E_U^{h,k} = (\mathcal{H}_U^k \cdot (\mathcal{H}_U^k)^\top) \cdot E_U^{b,k}, \tag{13}$$

$$E_I^{h,k} = (\mathcal{H}_I^k \cdot (\mathcal{H}_I^k)^\top) \cdot E_I^{b,k}, \tag{14}$$

where $(\mathcal{H}_U^k \cdot (\mathcal{H}_U^k)^\top) \in \mathbb{R}^{M \times M}$ and $(\mathcal{H}_I^k \cdot (\mathcal{H}_I^k)^\top) \in \mathbb{R}^{N \times N}$ correspond to the user-side and item-side hypergraph dependency matrices, respectively.

After hypergraph learning in the k-th behavior, the hypergraph embeddings of user u and item i are represented as $e_u^{h,k} \in \mathbb{R}^d$ and $e_i^{h,k} \in \mathbb{R}^d$, respectively. Notably, the hypergraph learning proposed in this study distinguishes itself from previous studies [36]. First, to incorporate more user interaction information during the hypergraph convolution, we inject the learned embeddings from the behavior interaction graph into the process of hypergraph learning. Second, to preserve the independence of behavior interaction graph learning, we adopt a stop-gradient strategy to ensure unidirectional information propagation from behavior-specific interaction graph learning to behavior-specific hypergraph learning. Third, to simplify the process of information propagation in the hypergraph learning, we remove activation functions from the hypergraph convolution.

4.2.3. Behavior-specific embedding integration

With the learned embeddings of users and items from the k-th behavior-specific interaction graph and behavior-specific hypergraph, we integrate them to obtain their corresponding behavior-specific embeddings:

$$e_u^k = e_u^{b,k} \oplus e_u^{h,k} \oplus e_u^{k-1}, \tag{15}$$

$$e_i^k = e_i^{b,k} \oplus e_i^{h,k} \oplus e_i^{k-1}, \tag{16}$$

where e_u^{k-1} and e_i^{k-1} are residual-connected embeddings derived from the previous behavior, e_u^k and e_i^k denote the updated embeddings for user u and item i from the k-th behavior, which will be utilized as initialization embeddings for the subsequent behavior. Note that we initialize the embeddings of user u and item i for the first behavior with the corresponding global embedding representations e_u^g and e_i^g, respectively.

4.3. Behavior consistency-guided contrastive learning

Although the cascading architecture effectively captures dependencies between consecutive behaviors, it falls short in fully exploring users' shared interest preferences across different behaviors and their multi-granularity interest preferences within behavior-specific interaction graphs and hypergraphs. To address this, we introduce inter- and intra-behavior consistency constraints through a contrastive learning framework, enabling a more comprehensive modeling of users' interest preferences from multiple perspectives.

4.3.1. Inter-behavior consistency contrastive learning

Given the variations in behavioral patterns, directly constraining the embedding consistency of users or items across different behaviors may hinder the exploration of common information across them. Consequently, we employ the global graph as an intermediary and align embeddings of nodes learned from each behavior graph towards those from the global graph. Specifically, we enhance the similarity of embeddings for the same node under both the global graph $\mathcal{G}_g$ and the k-th behavior graph $\mathcal{G}_k$ (i.e., positive sample pairs $(e_u^g, e_u^{b,k})$), while the embeddings for different users are encouraged to diverge from each other (i.e., negative sample pairs $(e_u^g, e_v^{b,k})$). Similar to InfoNCE [27], we formalize the inter-behavior contrastive loss $\mathcal{L}_{cl,user}^{g,b}$ between the global graph and all K behavior-specific interaction graphs on the user-side as follows:

$$\mathcal{L}_{cl,user}^{g,b} = \sum_{k=1}^{K} \sum_{u\in\mathcal{U}} -\log \frac{\exp(\phi(e_u^g, e_u^{b,k})/\tau)}{\sum_{v\in\mathcal{U}} \exp(\phi(e_u^g, e_v^{b,k})/\tau)}, \tag{17}$$

where τ denotes the temperature coefficient, and $\phi(\cdot)$ is the similarity function. The computation of $\mathcal{L}_{cl,item}^{g,b}$ on the item-side adopts an analogous approach. The comprehensive contrastive loss $\mathcal{L}_{cl}^{g,b}$ is obtained by combining the user-side and the item-side losses:

$$\mathcal{L}_{cl}^{g,b} = \mathcal{L}_{cl,user}^{g,b} + \mathcal{L}_{cl,item}^{g,b}. \tag{18}$$

Similarly, we obtain the inter-behavior contrastive loss $\mathcal{L}_{cl}^{g,h}$ between the global graph and all K behavior-specific hypergraphs as follows:

$$\mathcal{L}_{cl}^{g,h} = \mathcal{L}_{cl,user}^{g,h} + \mathcal{L}_{cl,item}^{g,h}, \tag{19}$$

where $\mathcal{L}_{cl,user}^{g,h}$ and $\mathcal{L}_{cl,item}^{g,h}$ represent the contrastive losses on the user-side and item-side, respectively.

4.3.2. Intra-behavior consistency contrastive learning

The intrinsic consistency between every user-item interaction graph and its corresponding hypergraph is maintained through our specially designed intra-behavior contrastive learning approach, which preserves mutual dependencies across both graph representations. Specifically, the intra-behavior contrastive loss for the user-side can be formally expressed as follows:

$$\mathcal{L}_{cl,user}^{b,h} = \sum_{k=1}^{K} \sum_{u\in\mathcal{U}} -\log \frac{\exp(\phi(e_u^{b,k}, e_u^{h,k})/\tau)}{\sum_{v\in\mathcal{U}} \exp(\phi(e_u^{b,k}, e_v^{h,k})/\tau)}, \tag{20}$$

where $(e_u^{b,k}, e_u^{h,k})$ denote the positive sample pairs under the k-th behavior and $(e_u^{b,k}, e_v^{h,k})$ are the negative sample pairs. The calculation of intra-behavior contrastive loss on the item-side $L_{cl,item}^{b,h}$ follows a similar process. Finally, the overall intra-behavior contrastive loss is defined as follows:

$$\mathcal{L}_{cl}^{b,h} = \mathcal{L}_{cl,user}^{b,h} + \mathcal{L}_{cl,item}^{b,h}. \tag{21}$$

4.4. Model prediction

We propose a behavior mutual enhancement module to refine current behavior by extracting and integrating information from other behaviors. For the k-th behavior, we update the representations of users and items as follows:

$$\begin{aligned} \tilde{e}_u^k &= \text{softmax}\left(\frac{e_u^k \cdot E_u^b}{\sqrt{d}}\right) \cdot (E_u^b)^\top, \\ \tilde{e}_i^k &= \text{softmax}\left(\frac{e_i^k \cdot E_i^b}{\sqrt{d}}\right) \cdot (E_i^b)^\top, \end{aligned} \tag{22}$$

where $E_u^b = [e_u^1, e_u^2, \dots, e_u^K] \in \mathbb{R}^{d\times K}$, $E_i^b = [e_i^1, e_i^2, \dots, e_i^K] \in \mathbb{R}^{d\times K}$.

To obtain comprehensive representations from both global and local perspectives, we integrate the k-th behavior-specific user/item embeddings with their corresponding global user/item embeddings as follows:

$$\bar{e}_u^k = \tilde{e}_u^k \oplus e_u^g, \tag{23}$$

$$\bar{e}_i^k = \tilde{e}_i^k \oplus e_i^g. \tag{24}$$

The ultimate prediction score y_{ui}^k indicates the engagement probability for user-item pair (u, i) under k-behavior, formulated as:

$$y_{ui}^k = (\bar{e}_u^k)^\top \cdot \bar{e}_i^k. \tag{25}$$

4.5. Joint optimization

In the training phase, we employ a joint learning framework based on multi-task learning [38] to explore the information across multiple behaviors. Specifically, each behavior is handled as a distinct prediction task, with the BPR loss [39] serving as the optimization target defined in the following manner:

$$\mathcal{L}_k = \sum_{(u,i,j)\in\mathcal{T}_k} -\ln \sigma(y_{ui}^k - y_{uj}^k), \tag{26}$$

where $\mathcal{T}_k = \{(u,i,j) \mid (u,i) \in \mathcal{R}^+, (u,j) \in \mathcal{R}^-\}$ indicates the positive and negative samples in the training set, $\mathcal{R}^+$ ($\mathcal{R}^-$) denotes the sample set with observed (unobserved) interaction in the k-th behavior.

Accordingly, we formulate the loss function $\mathcal{L}$ for the complete task as:

$$\mathcal{L} = \underbrace{\sum_{k=1}^{K} \mathcal{L}_k}_{prediction} + \alpha(\underbrace{\lambda_1 \mathcal{L}_{cl}^{g,b} + \lambda_2 \mathcal{L}_{cl}^{g,h}}_{inter-behavior} + \underbrace{\lambda_3 \mathcal{L}_{cl}^{b,h}}_{intra-behavior}) + \beta\|\Theta\|_2, \tag{27}$$

where α is the coefficient of the overall consistency loss, β denotes the weighting factor for regularization, Θ stands for all learnable parameters, and λ_1, λ_2 and λ_3 are the coefficients of three individual consistency losses, respectively.

4.6. Complexity analysis

We analyze the computational complexity of CMC-GCN's key components. For the global graph learning component, the computational complexity is $\mathcal{O}(L \times |\mathcal{G}_g| \times d)$, where $|\mathcal{G}_g|$ denotes the number of edges in the global graph $\mathcal{G}_g$. The multi-granularity cascading graph convolution module comprises two parts: (1) Behavior-specific interaction graph learning has a computational complexity of $\mathcal{O}(\sum_{k=1}^{K} L\times|\mathcal{G}_k|\times d)$, with $|\mathcal{G}_k|$ being the edge count of the k-th behavior graph. (2) Behavior-specific hypergraph learning, utilizing low-rank implementation and optimized computation order (e.g., the computation $E_U^{h,k} = (\mathcal{H}_U^k \cdot (\mathcal{H}_U^k)^\top) \cdot E_U^{b,k}$ can be optimized as $E_U^{h,k} = \mathcal{H}_U^k \cdot ((\mathcal{H}_U^k)^\top \cdot E_U^{b,k})$), has complexity $\mathcal{O}(K \times L \times (M + N) \times S \times d)$. The behavior-consistency guided contrastive learning module operates at batch-level with computational complexity $\mathcal{O}(K \times B \times (M + N) \times d)$, where B is the batch size.

5. Experiments

5.1. Experimental settings

5.1.1. Datasets

Consistent with prior research [13,16,17], we evaluate our approach on four real-world benchmark datasets: Beibei, Taobao, Tmall, and QKV. The complete statistical breakdown appears in Table 1.

Beibei[1] serves as a common benchmark for multi-behavior recommendation tasks. The underlying data originates from its namesake platform - China's premier infant product e-commerce platform, which holds a huge market share in its specialized domain.

Taobao[2] dataset originates from its namesake e-commerce platform, a dominant force in China's online retail sector. The collection spans 11,953 items and 15,449 users.

Tmall[3] is built from consumption records of its platform, a leading Chinese e-commerce hub. The curated dataset includes interactions with 11,953 merchandise items and 41,738 distinct users.

QKV is constructed using the QK-Video data from Tenrec [40], which comprises various user interaction records during video browsing, with the "share" action designated as the target behavior.

5.1.2. Baselines

The CMC-GCN framework developed in this study was evaluated through comparative experiments against two categories of recommendation methods.

Table 1
Overview of experimental datasets.

Dataset	Users	Items	Interaction	Behavior type
Beibei	21,716	7977	3,338,068	{View, Cart, Buy}
Taobao	15,449	11,953	1,161,610	{View, Cart, Buy}
Tmall	41,738	11,953	2,292,594	{View, Collect, Cart, Buy}
QKV	11,870	18,074	464,213	{Click, Like, Share}

[1] https://www.beibei.com/.
[2] https://www.taobao.com/.
[3] https://www.tmall.com/.

Single-behavior recommendation methods:

- **MF-BPR** [39]. It introduces BPR-Opt, a general-purpose training objective specifically designed for matrix factorization architectures, which employs both observed and unobserved interaction samples through a maximum posterior estimation paradigm.
- **NeuMF** [41]. This neural recommendation framework employs a hybrid architecture that combines GMF with MLP components to capture complex user-item relationships through collaborative filtering.
- **LightGCN** [1]. This GCN-driven method focuses on capturing user and item characteristics by propagating information through a bipartite interaction structure.

Multi-behavior recommendation methods:

- **RGCN** [42]. RGCN is designed to handle graph learning tasks with multiple types of edges, enabling the propagation of type-specific graph information by distinguishing various types of edges.
- **GNMR** [11]. This method employs heterogeneous graph neural networks to model collaborative signals across diverse categories of interactions among users and items, while capturing behavior dependency through recursive embedding propagation.
- **NMTR** [15]. It simultaneously models user preferences under different user behaviors by using cascaded prediction, and adopts a multi-task optimization framework.
- **MBGCN** [10]. MBGCN leverages a unified multi-behavior graph that quantifies behavioral influences across interaction types for target prediction, and introduces an item–item graph to extract richer interaction information.
- **S-MBRec** [12]. This model introduces a supervised task that distinguishes the significance of various behaviors and captures commonality through a target behavior-centered contrastive learning framework.
- **CRGCN** [18]. CRGCN explicitly exploits the dependencies among sequential behaviors and develops a cascading residual module to gradually learn and refine user preferences.
- **MB-CGCN** [16]. MB-CGCN explicitly models dependencies between behaviors by introducing a feature transformation module in the chaining pass of behaviors.
- **MB-HGCN** [13]. This model utilizes a hierarchical graph convolutional network that generates representation vectors for both users and items from a global perspective to local behavior-specific perspective to jointly model user preferences.
- **PKEF** [17]. PKEF leverages a parallel knowledge architecture to strengthen the hierarchical information propagation for mitigating skewed interaction distribution across varied behavioral patterns. Moreover, it incorporates a projection-based decomposition approach to distinguish between shared and distinctive feature representations among different behaviors.
- **BCIPM** [14]. BCIPM introduces a behavior-aware preference modeling framework that contextualizes item preferences based on interaction patterns across different behavior types.
- **DeMBR** [32]. DeMBR eliminates hard noise via pruning-based denoising with a memory bank and suppresses soft noise through semantic guidance from strongly expressive behaviors, thereby more accurately capturing user preferences.

5.1.3. Implementation details

To ensure fairness in comparison, the configuration settings from [17] were adopted, which adopt an embedding size from {64, 128}. The learning rate is initialized to $5e^{-4}$, and the regularization coefficient β is set to $1e^{-3}$. We utilize Adam [43] to optimize the model parameters. For the number of GCN layers in each behavior, we consider options among {1, 2, 3}. The coefficient α is tuned within {0.1, 0.5}. Additionally, coefficients λ_1, λ_2, and λ_3 are adjusted in {0, 0.5, 1.0, 1.5, 2.0, 2.5}, with the constraint that their sum equals 3. The temperature coefficient τ

Table 2
Performance of CMC-GCN compared to baseline as measured using HR@10 and NDCG@10. The top two results for each column are highlighted in bold and underlined, respectively. ('Rel Impr.' indicates the percentage gain achieved by CMC-GCN in comparison with the most competitive baseline.)

Type	Method	Beibei		Taobao		Tmall		QKV	
		HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
Single-behavior	MF-BPR	0.0284	0.0142	0.0128	0.0059	0.0154	0.0072	0.0287	0.0154
	NeuMF	0.0231	0.0124	0.0077	0.0035	0.0100	0.0049	0.0223	0.0115
	LightGCN	0.0318	0.0174	0.0381	0.0224	0.0450	0.0249	0.0327	0.0163
Multi-behavior	RGCN	0.0363	0.0188	0.0215	0.0104	0.0316	0.0157	0.0755	0.0435
	GNMR	0.0413	0.0221	0.0368	0.0216	0.0393	0.0193	0.0971	0.0557
	NMTR	0.0429	0.0198	0.0282	0.0137	0.0536	0.0286	0.1153	0.0664
	MBGCN	0.0470	0.0259	0.0509	0.0294	0.0549	0.0285	0.0991	0.0542
	S-MBRec	0.0489	0.0253	0.0662	0.0353	0.0694	0.0362	0.1169	0.0704
	CRGCN	0.0459	0.0324	0.0855	0.0439	0.0840	0.0442	0.1783	0.1004
	MB-CGCN	0.0579	0.0381	0.1233	0.0677	0.0984	0.0558	0.1805	0.1033
	MB-HGCN	0.0744	0.0369	0.1380	0.0728	0.1470	0.0784	0.2070	0.1266
	PKEF	<u>0.1130</u>	0.0582	0.1385	0.0785	0.1277	0.0721	0.1456	0.0729
	BCIPM	0.0816	0.0406	<u>0.1426</u>	<u>0.0807</u>	<u>0.1512</u>	0.0803	0.2239	0.1359
	DeMBR	0.1046	<u>0.0585</u>	0.1392	0.0806	0.1458	<u>0.0840</u>	<u>0.2440</u>	<u>0.1416</u>
	CMC-GCN	**0.1347**	**0.0675**	**0.1960**	**0.1061**	**0.1715**	**0.0922**	**0.2771**	**0.1610**
	Rel Impr.	19.20 %	15.38 %	37.45 %	31.47 %	13.43 %	9.76 %	13.57 %	13.70 %

is fixed at 0.1. The number of hyperedges S is tuned in $\{32, 64\}$. The optimal settings for the consistency constraints λ_1, λ_2, and λ_3, the temperature coefficient τ, and the number of hyperedges S are discussed in Section 5.9.

5.1.4. Evaluation metrics

Following [1,41,44], our evaluation adopts the leave-one-out strategy: (1) the test set comprises users' latest interactions, while (2) the validation set utilizes their penultimate interactions. To assess the effectiveness of our model, we adopt Hit Ratio (HR@n) and Normalized Discounted Cumulative Gain (NDCG@n) as evaluation metrics.

5.2. Performance evaluation

Table 2 displays the measured performance metrics in our comparative study. The analysis reveals these key findings: First, among all approaches focusing on single-behavior modeling, LightGCN consistently outperforms MF-BPR and NeuMF across all datasets. The reason for the performance improvement is that LightGCN effectively captures high-order interaction information, whereas both MF-BPR and NeuMF focus solely on information from direct neighbors. Second, among all multi-behavior methods, RGCN shows the lowest performance across all cases. This is because it combines information from different behaviors through simple summation, which overlooks the unique characteristics of each behavior. GNMR achieves better performance than RGCN by employing heterogeneous graph neural networks, which can distinguish the unique characteristics of different behaviors. NMTR obtains a competitive performance, similar to GNMR and also outperforms RGCN since it explores the sequential dependency between behaviors by indirectly modeling the interaction scores associated with each behavior.

MBGCN outperforms both GNMR and NMTR as it further captures the contribution degree of different behaviors and assigns specific weights to them. Compared to MBGCN, S-MBRec demonstrates enhanced performance through the introduction of a supervised learning objective that more effectively captures the significance of various interaction types. CRGCN and MB-CGCN achieve superior performance by leveraging cascading graph convolutional networks to capture the cascade relationships among multiple behaviors. MB-HGCN yields a better performance than CRGCN and MB-CGCN. The reason is that MB-HGCN effectively leverages multi-behavior information through its hierarchical learning and aggregation strategies. PKEF outperforms MB-HGCN as it can learn complex interactions across multiple behaviors by integrating both cascade and parallel paradigms. BCIPM demonstrates strong performance, particularly in capturing nuanced item-specific preferences within specific behaviors via its behavior-contextualized item preference network. Relative to other baseline methods, DeMBR demonstrates superior performance across the majority of experimental scenarios, attributed to its soft and hard denoising mechanisms, which jointly optimize cross-behavior semantics.

By a wide margin, our proposed method CMC-GCN surpasses the best performing baseline. For instance, CMC-GCN achieves performance gains of 19.20 %, 37.45 %, 13.43 % and 13.57 % in HR@10 compared to the strongest baseline across Beibei, Taobao, Tmall and QKV datasets respectively. This is because CMC-GCN captures both fine- and coarse-grained hierarchical correlations by incorporating the user-item interaction graph along with a behavior-specific hypergraph. Additionally, it ensures inter- and intra-behavior consistency in representations through a contrastive learning framework.

5.3. Ablation study

To further elaborate on reasons for the performance gain and validate the design rationalization of key components in CMC-GCN, we introduce additional variants as follows:

- $w/o\ global$: We remove the global graph as well as the cross-behavior contrastive loss associated with it.
- $w/o\ hyper$: We discard behavior-specific hypergraphs from the multi-granularity cascading graph convolution network module.
- $w/o\ stop$: The stop-gradient strategy between the interaction graph and the hypergraph across behaviors is removed.
- $w/o\ cascading$: The cascading architecture is eliminated, and embeddings for multiple behaviors are all initialized based on the global graph.
- $w/o\ mutual$: We remove the behavior mutual enhancement module, and directly adopt embeddings learned from each behavior for prediction.

As presented in Table 3, it is evident that all components play important roles in improving the effectiveness of our method. To be specific, we observe that: First, removing the global graph ($w/o\ global$) leads to a significant performance degradation. This is attributable to its essential function in capturing the user's overall preferences across various behaviors. Moreover, it acts as a mediator for cross-behavior contrastive learning, which imposes an inherent constraint to maintain consistency across different behaviors. Second, abandoning hypergraphs ($w/o\ hyper$) causes a notable performance deterioration, highlighting the importance of incorporating learnable hypergraphs. This result is attributed to the hypergraph's ability to capture implicit high-order correlations among users or items, thereby complementing the behavior interaction graph. Third, discarding the stop-gradient strategy causes a performance drop in the proposed model, which reveals the importance

Table 3
Performance comparison of different CMC-GCN variants on all datasets.

Methods	Beibei		Taobao		Tmall		QKV	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
CMC-GCN(full)	**0.1347**	**0.0675**	**0.1960**	**0.1061**	**0.1715**	**0.0922**	**0.2771**	**0.1610**
w/o $global$	0.0664	0.0342	0.1493	0.0861	0.0678	0.0359	0.1913	0.1206
w/o $hyper$	0.1220	0.0616	0.1699	0.0910	0.1607	0.0861	0.2410	0.1417
w/o $stop$	0.1157	0.0592	0.1820	0.0978	0.1714	0.0921	0.2630	0.1550
w/o $cascading$	0.1243	0.0630	0.1766	0.0949	0.1633	0.0873	0.2544	0.1489
w/o $mutual$	0.1076	0.0550	0.1768	0.0946	0.1585	0.0860	0.2280	0.1362

The bold values indicate the best performing results.

Table 4
Performance comparison of different contrastive learning variants for CMC-GCN on all datasets.

Methods	Beibei		Taobao		Tmall		QKV	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
CMC-GCN (full)	**0.1347**	**0.0675**	**0.1960**	**0.1061**	**0.1715**	**0.0922**	**0.2771**	**0.1610**
w/o CL_{intra}	0.1200	0.0605	0.1886	0.1013	0.1681	0.0902	0.2664	0.1544
w/o CL_{inter}	0.0970	0.0498	0.1671	0.0877	0.1531	0.0794	0.2601	0.1516
w/o CL_{all}	0.0854	0.0438	0.1343	0.0709	0.1400	0.0745	0.2093	0.1272

The bold values indicate the best performing results.

of maintaining the unidirectional information propagation from fine-grained interaction graph learning to coarse-grained behavior-specific hypergraph learning. Fourth, the results of the variant w/o $cascading$ validate the benefits of modeling the dependencies among sequential behaviors, which is consistent with previous work. Finally, the variant w/o $mutual$, which excludes behavior mutual enhancement, shows a significant performance decline, highlighting the effectiveness of integrating information from other behaviors.

5.4. Effectiveness analysis of behavior consistency-guided contrastive learning

To investigate the contribution of the inter- and intra-behavior consistency constraints based on contrastive learning, we conduct experiments by removing the constraints from CMC-GCN and obtain the following variants:

- w/o CL_{intra}: We discard the consistency constraint between each user-item interaction graph and its corresponding hypergraph in the same behavior, i.e., the intra-behavior contrastive loss $\mathcal{L}_{cl}^{b,h}$ is removed.
- w/o CL_{inter}: We remove the consistency constraint across different behaviors, i.e., the inter-behavior contrastive loss ($\mathcal{L}_{cl}^{g,b}$ and $\mathcal{L}_{cl}^{g,h}$) is ignored.
- w/o CL_{all}: Both intra- and inter-behavior contrastive losses are simultaneously removed.

As shown in Table 4, our findings demonstrate that both intra- and inter-behavior contrastive constraints play a crucial role in CMC-GCN, and removing the inter-behavior contrastive constraint results in a greater decline in model performance than removing the intra-behavior contrastive constraint on all datasets. Furthermore, discarding both constraints leads to an even more severe performance degradation, indicating that the intra- and inter-behavior contrastive constraints can complement each other.

5.5. Model performance under different interaction densities

To assess CMC-GCN's performance at varying interaction densities, the test set users are categorized into five distinct categories according to their interaction frequency within the Taobao dataset. Specifically, we separate users into five categories: G1 (1–3 interactions), G2 (4–6 interactions), G3 (7–9 interactions), G4 (10–12 interactions) and G5 (over 12 interactions). We illustrate our proposed model's performance for each category in Fig. 2.

In comparison with the four most advanced baselines (e.g., MB-HGCN, PKEF, BCIPM and DeMBR), our method consistently demonstrates the optimal recommendation performance across all categories. Furthermore, the CMC-GCN's performance gain over these strong baselines grows as the interaction density levels increase. The experimental outcomes affirm the efficacy of incorporating learnable hypergraphs combined with intra- and inter-behavior contrastive constraints within our proposed CMC-GCN.

5.6. Model performance in cold-start scenario

This section examines the efficacy of CMC-GCN in addressing cold-start challenges. To analyze how different cold-start degrees affect the proposed model's performance, we implement controlled experiments on the Taobao dataset by randomly selecting 50 % of users and systematically removing portions of their interaction history (10 %, 20 %, 30 %, 40 %, and 50 % deletion ratios). We compare CMC-GCN with MB-HGCN, PKEF, BCIPM and DeMBR. Fig. 3 shows the relative performance degradation of different methods in terms of HR@10 compared to the performance on full interaction data. Our approach demonstrates less performance degradation compared to other baselines. This is primarily due to the following reasons: (1) The behavior-specific hypergraph in CMC-GCN is effective in capturing complex high-order correlations among users and items, which complements its corresponding behavior-specific interaction graph to mitigate the cold-start issue. (2) CMC-GCN incorporates the inter-behavior consistency constraint which can further strengthen the representation learning in each behavior by incorporating information from other behaviors.

5.7. Impact of auxiliary behaviors

To examine how auxiliary behaviors affect model performance, we gradually remove these behaviors and assess the CMC-GCN's performance alongside four competitive baselines: MB-HGCN, PKEF, BCIPM and DeMBR. Fig. 4 demonstrates the experimental outcomes using the Taobao dataset.

Our findings indicate that eliminating the auxiliary behavior “view” causes a notable degradation in performance across all methods. In addition, a further performance drop can be observed if we continue to discard the auxiliary behavior “cart”. Experimental findings reveal the beneficial effects of integrating additional auxiliary behaviors across different methods, where our proposed approach demonstrates superior utilization efficiency of these behaviors. Moreover, among different settings of auxiliary behaviors, CMC-GCN consistently outperforms all four

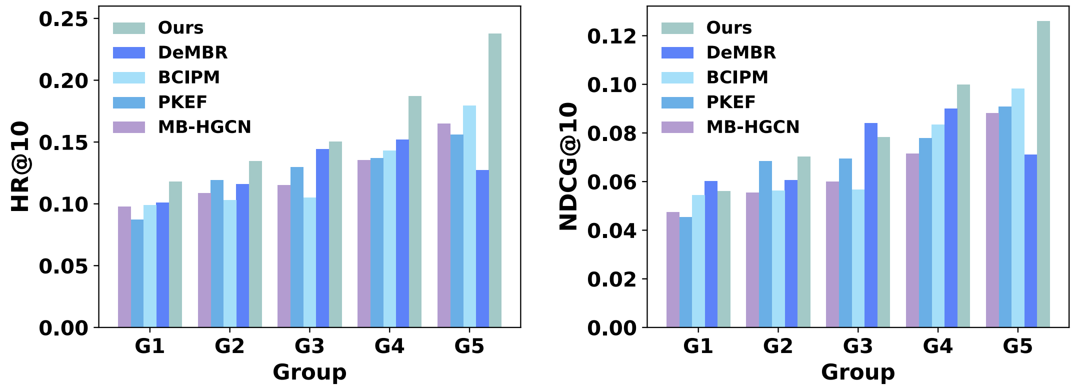


Fig. 2. The performance of our method versus baseline methods at different interaction densities.

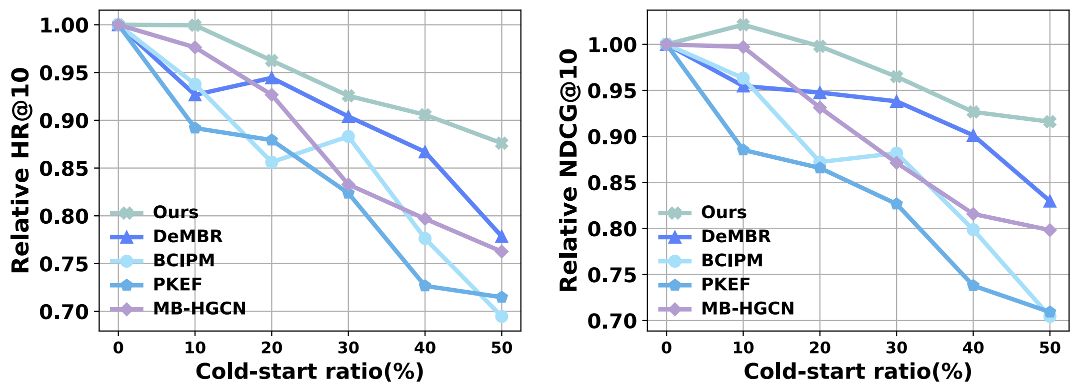


Fig. 3. The impact of different cold-start ratios on model performance.

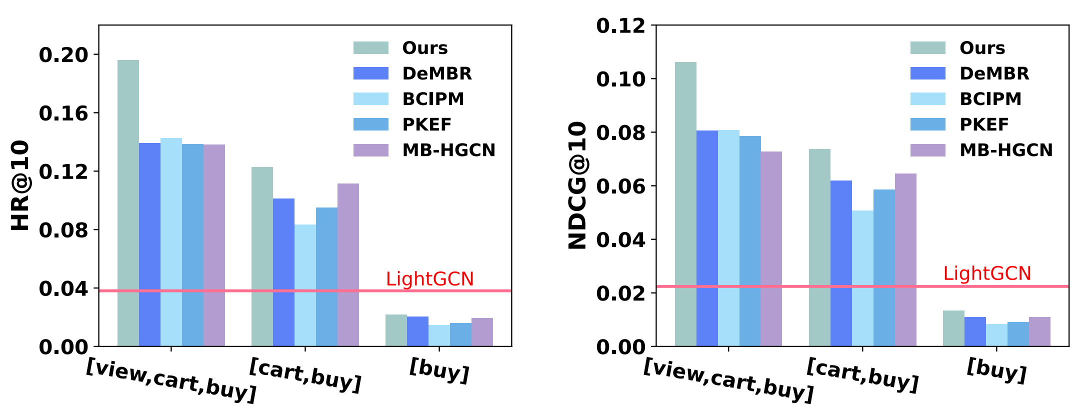


Fig. 4. The Impact of auxiliary behaviors on performance.

strong baselines, which suggests the better capability of CMC-GCN for effectively exploring multi-behavior information. Additionally, when only utilizing the target behavior "buy", all recommendation models that incorporate multiple behaviors, including our proposed CMC-GCN, are lower than a single-behavior recommendation model like LightGCN [1]. This may be ascribed to the intricate architecture of such multi-behavior methods that could exhibit enhanced susceptibility to overfitting.

5.8. Impact of behavior chain order

Within a sequence of behaviors, subsequent behaviors typically provide a more pronounced reflection of user preferences compared to earlier ones [16], where we refer to the earlier behavior as causal behavior. To investigate the impact of sequential behavior patterns within the framework on CMC-GCN's effectiveness, we adjust the order of behaviors in the Taobao dataset, and the corresponding outcomes are displayed in Fig. 5.

As shown, the original behavior chain [view → cart → buy] results in the best performance, whereas its reverse order yields the worst performance, even worse than the variant without a cascading architecture (i.e., variant $w/o\ \ cascading$). In addition, placing more casual behaviors later in a behavior chain leads to a performance decline of our proposed model. For example, the model performance on the behavior chain [buy → view → cart] is inferior to that on the behavior chain [view → buy → cart]. This result indicates that the sequence of behaviors serves as a critical determinant in the effectiveness of CMC-GCN. The primary reason lies in the fact that casual behaviors, such as 'view',

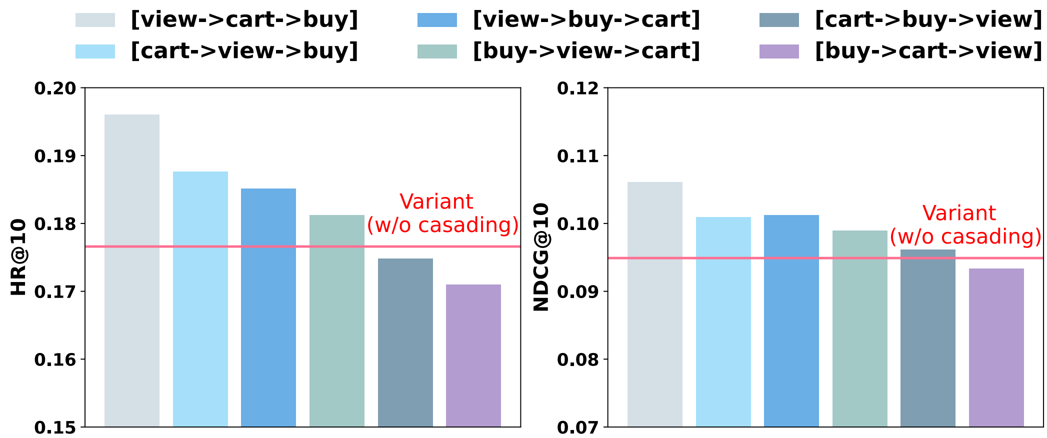


Fig. 5. Impact of behavior sequence order on performance in the cascading architecture.

Table 5
Performance differences when assigning varying weights to consistency constraint terms in training.

$[\lambda_1, \lambda_2, \lambda_3]$	Beibei		Taobao		Tmall		QKV	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
[0.5, 0.5, 2.0]	0.1173	0.0591	0.1804	0.0964	0.1615	0.0877	0.2673	0.1561
[0.5, 2.0, 0.5]	**0.1347**	**0.0675**	0.1941	0.1032	0.1634	0.0884	0.2695	0.1571
[1.0, 1.0, 1.0]	0.1249	0.0631	0.1888	0.1001	0.1574	0.0856	0.2566	0.1499
[1.0, 0.5, 1.5]	0.1287	0.0653	0.1814	0.0970	0.1680	0.0914	0.2540	0.1493
[1.5, 0.5, 1.0]	0.1310	0.0662	0.1838	0.0978	0.1686	0.0914	**0.2771**	**0.1610**
[1.5, 1.0, 0.5]	0.1293	0.0658	**0.1960**	**0.1061**	**0.1715**	**0.0922**	0.2737	0.1600
[2.0, 0.5, 0.5]	0.1266	0.0644	0.1937	0.1036	0.1665	0.0906	0.2657	0.1558

The bold values indicate the best performing results.

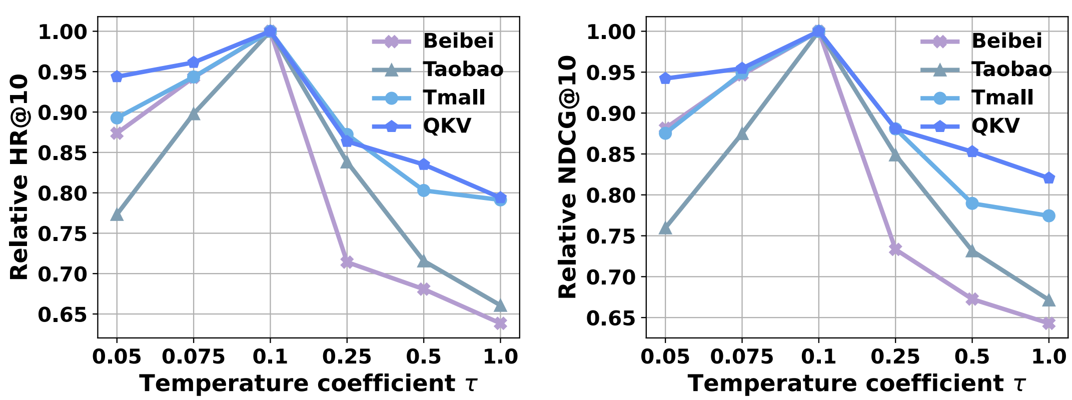


Fig. 6. The Impact of temperature coefficient τ.

often contain more noise. By positioning these casual behaviors earlier in the sequence, the cascading architecture can progressively refine the learning of user preferences, allowing a more effective transfer of user preferences from auxiliary behavior to the target behavior.

5.9. Impact of hyperparameter setting

To investigate the impact of different consistency weight combinations (λ_1, λ_2, and λ_3) on model performance across datasets, we tested various configurations without exhaustively enumerating all possible combinations. The experimental results are shown in Table 5. From the tested combinations, we observed that no single weight configuration achieves optimal performance across all datasets. This indicates that while introducing inter-behavior and intra-behavior consistency constraints positively impacts model performance, weight adjustment remains necessary to account for data and scenario variations. We leave the study of automatically balancing behavior consistency constraints for future research.

The hyperparameter τ controls the smoothness of embedding similarity, i.e., a small τ results in sharper similarity, while a larger τ produces a smoother similarity. We tune τ from the set of values {0.05, 0.075, 0.1, 0.25, 0.5, 1.0} and present the corresponding findings for four datasets in Fig. 6. The experimental outcomes demonstrate that as τ increases, the performance of CMC-GCN initially exhibits a steady improvement, peaking at $\tau = 0.1$, followed by a decline with further augmentation of τ.

The hyperparameter S indicates the number of hyperedges utilized in a behavior-specific hypergraph. Each hyperedge serves as a mediator to connect users (items), and reflects a specific semantic dependency between users (items). We investigate the effect that the number of hyperedges has on the performance of CMC-GCN. To be specific, we vary the value of S in {16, 32, 64, 128, 256}, with the corresponding

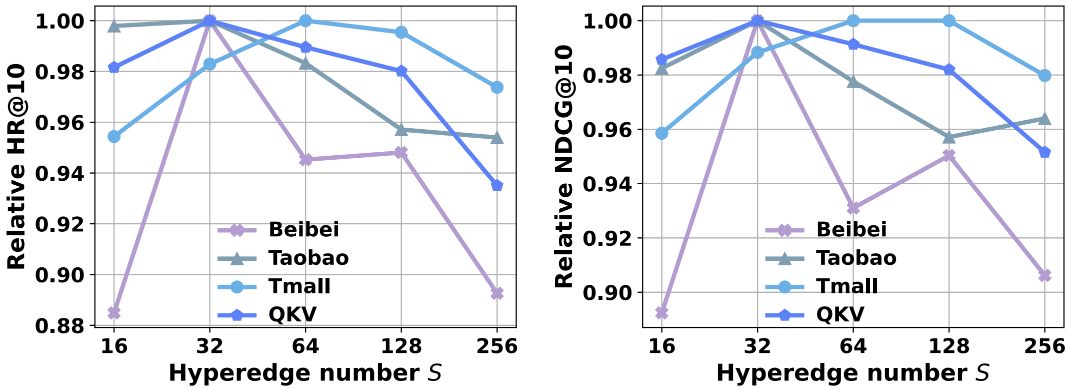


Fig. 7. The Impact of hyperedge number S.

Table 6
Model computational cost with running time (seconds).

Methods	Beibei		Taobao		Tmall		QKV	
	Training	Inference	Training	Inference	Training	Inference	Training	Inference
MB-HGCN	6.9	6.5	4.6	4.2	13.1	9.2	5.4	3.5
PKEF	40.1	13.9	36.7	10.5	54.4	15.6	25.1	9.1
BCIPM	217.5	27.0	280.3	14.0	431.5	32.7	111.8	18.1
DeMBR	65.3	7.7	24.6	5.3	271.7	23.1	8.4	5.2
CMC-GCN	8.9	7.5	8.5	5.1	15.2	9.7	7.0	4.5

outcomes presented in Fig. 7. Note that when S is small, CMC-GCN performs poorly. This could be due to the insufficient number of hyperedges, which limits the model's ability to adequately represent the complex semantic dependencies among users (items). As S gets larger, the model performance improves progressively. Model performance begins to degrade as we gradually increase the S parameter beyond a certain threshold. This phenomenon occurs because excessive values of S may suffer from over-fitting issues and harm model performance.

5.10. Model training and inference efficiency study

To evaluate training and inference efficiency, we compare the per-epoch time consumption of CMC-GCN with other multi-behavior methods under same experimental conditions, as shown in Table 6. Specifically, MB-HGCN achieves superior efficiency due to its simple hierarchical architecture and reliance solely on multi-behavior graph convolutional networks. PKEF incurs additional computational costs from its parallel knowledge enhancement and multi-expert network, while BCIPM suffers from prolonged training time caused by its complex yet essential embedding pre-training operations. DeMBR's semantic guidance denoising requires pairwise behavior difference computation, leading to high time costs. As behavior types grow (e.g., Tmall's 4 behaviors), the combinatorial explosion of behavior pairs significantly increases computational complexity. While our method introduces additional computational costs by incorporating behavior-specific hypergraphs within each behavior and implementing inter- and intra-behavior consistency contrastive learning, the low-rank implementation of hypergraphs maintains overall computational efficiency. Consequently, our approach achieves efficiency comparable to the optimal MB-HGCN while outperforming other methods.

6. Conclusion

In this paper, we present a novel model CMC-GCN for multi-behavior recommendation. CMC-GCN examines both fine- and coarse-grained correlations among users or items for each behavior by jointly modeling the behavior-specific interaction graph and its corresponding hypergraph, where the latter is designed to capture coarse-grained correlations among users or items, helping to mitigate the sparsity issues inherent in the interaction graph. Additionally, it develops a behavior consistency-guided alignment strategy that ensures representation consistency between the interaction graph and its corresponding hypergraph within each behavior, while also maintaining consistency across different behaviors. Extensive experiments on four real-world benchmark datasets demonstrate that our proposed method considerably outperforms state-of-the-art baseline methods.

CRediT authorship contribution statement

Yabo Yin: Writing – original draft, Software, Methodology, Conceptualization. **Xiaofei Zhu:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Conceptualization. **Kunyang Huang:** Software, Methodology. **Wenshan Wang:** Supervision, Methodology. **Yihao Zhang:** Supervision. **Pengfei Wang:** Supervision. **Yixing Fan:** Supervision. **Jiafeng Guo:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [grant number 62472059, 62372059]; the Natural Science Foundation of Chongqing, China [grant number CSTB2022NSCQ-MSX1672]; the Chongqing Talent Plan Project, China [grant number CSTC2024YCJH-BGZXM0022]; the Major Project of Science and Technology Research Program of Chongqing Education Commission of China [grant number KJZD-M202201102].

Data availability

The relevant code and data have been made public and links have been provided in the manuscript.

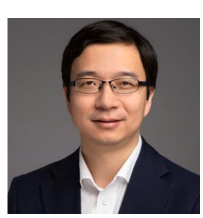
References

[1] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, LightGCN: simplifying and powering graph convolution network for recommendation, in: SIGIR, 2020, pp. 639–648.
[2] L. Liu, L. Wang, T. Lian, Case4sr: using category sequence graph to augment session-based recommendation, Knowl. Based Syst. 212 (2021) 106558.
[3] J. Qiao, L. Wang, Modeling user micro-behaviors and original interest via adaptive multi-attention network for session-based recommendation, Knowl. Based Syst. 244 (2022) 108567.
[4] L. Tang, B. Long, B. Chen, D. Agarwal, An empirical study on recommendation with multiple types of feedback, in: SIGKDD, 2016, pp. 283–292.
[5] Z. Zhao, Z. Cheng, L. Hong, E.H. Chi, Improving user topic interest profiles by behavior factorization, in: Proceedings of WWW, 2015, pp. 1406–1416.
[6] B. Loni, R. Pagano, M.A. Larson, A. Hanjalic, Bayesian personalized ranking with multi-channel user feedback, in: RecSys, 2016, pp. 361–364.
[7] J. Ding, G. Yu, X. He, Y. Quan, Y. Li, T. Chua, D. Jin, J. Yu, Improving implicit recommender systems with view data, in: IJCAI, 2018, pp. 3343–3349.
[8] L. Guo, L. Hua, R. Jia, B. Zhao, X. Wang, B. Cui, 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, 2019, pp. 1984–1992.
[9] C. Chen, M. Zhang, Y. Zhang, W. Ma, Y. Liu, S. Ma, Efficient heterogeneous collaborative filtering without negative sampling for recommendation, in: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020, pp. 19–26.
[10] B. Jin, C. Gao, X. He, D. Jin, Y. Li, Multi-behavior recommendation with graph convolutional networks, in: SIGIR, 2020, pp. 659–668.
[11] L. Xia, C. Huang, Y. Xu, P. Dai, M. Lu, L. Bo, Multi-behavior enhanced recommendation with cross-interaction collaborative relation modeling, in: Proceedings of the 37th IEEE International Conference on Data Engineering, 2021, pp. 1931–1936.
[12] S. Gu, X. Wang, C. Shi, D. Xiao, Self-supervised graph neural networks for multi-behavior recommendation, in: IJCAI, 2022, pp. 2052–2058.
[13] M. Yan, Z. Cheng, J. Sun, F. Sun, Y. Peng, MB-HGCN: a hierarchical graph convolutional network for multi-behavior recommendation, CoRR arXiv:2306.10679, (2023).
[14] M. Yan, F. Liu, J. Sun, F. Sun, Z. Cheng, Y. Han, 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, pp. 946–955.
[15] C. Gao, X. He, D. Gan, X. Chen, F. Feng, Y. Li, T. Chua, D. Jin, Neural multi-task recommendation from multi-behavior data, in: Proceedings of the 35th IEEE International Conference on Data Engineering, 2019, pp. 1554–1557.
[16] Z. Cheng, S. Han, F. Liu, L. Zhu, Z. Gao, Y. Peng, Multi-behavior recommendation with cascading graph convolution networks, in: WWW, 2023, pp. 1181–1189.
[17] C. Meng, C. Zhai, Y. Yang, H. Zhang, X. Li, Parallel knowledge enhancement based framework for multi-behavior recommendation, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023, pp. 1797–1806.
[18] M. Yan, Z. Cheng, C. Gao, J. Sun, F. Liu, F. Sun, H. Li, Cascading residual graph convolutional network for multi-behavior recommendation, CoRR arXiv:2205.13128, (2022).
[19] A.P. Singh, G.J. Gordon, Relational learning via collective matrix factorization, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 650–658.
[20] G. Guo, H. Qiu, Z. Tan, Y. Liu, J. Ma, X. Wang, Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems, Knowl. Based Syst. 138 (2017) 202–207.
[21] C. Huang, Recent advances in heterogeneous relation learning for recommendation, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, 2021, pp. 4442–4449.
[22] D. Wu, X. Luo, Z. Ma, C. Chen, M. Deng, J. Ma, ARGO: modeling heterogeneity in e-commerce recommendation, in: International Joint Conference on Neural Networks, 2021, pp. 1–8.
[23] L. Xia, C. Huang, Y. Xu, P. Dai, X. Zhang, H. Yang, J. Pei, L. Bo, Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation, in: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021, pp. 4486–4493.
[24] W. Guo, C. Meng, E. Yuan, Z. He, H. Guo, Y. Zhang, B. Chen, Y. Hu, R. Tang, X. Li, R. Zhang, Compressed interaction graph based framework for multi-behavior recommendation, in: Proceedings of the ACM Web Conference 2023, 2023, pp. 960–970.
[25] C. Meng, H. Zhang, W. Guo, H. Guo, H. Liu, Y. Zhang, H. Zheng, R. Tang, X. Li, R. Zhang, Hierarchical projection enhanced multi-behavior recommendation, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 4649–4660.
[26] C. Meng, Z. Zhao, W. Guo, Y. Zhang, H. Wu, C. Gao, D. Li, X. Li, R. Tang, Coarse-to-fine knowledge-enhanced multi-interest learning framework for multi-behavior recommendation, ACM Trans. Inf. Syst. 42 (1) (2024) :30:1-:30:27.
[27] A.V.D. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, CoRr arXiv:1807.03748, (2018).
[28] Y. Huang, Z. Li, Z. Chen, C. Zhang, H. Ma, Sentence salience contrastive learning for abstractive text summarization, Neurocomputing 593 (2024) 127808.
[29] Y. Zhang, Y. Zhang, W. Liao, X. Li, X. Wang, Multi-view self-supervised learning on heterogeneous graphs for recommendation, Appl. Soft Comput. 174 (2025) 113056.
[30] H. Xuan, Y. Liu, B. Li, H. Yin, Knowledge enhancement for contrastive multi-behavior recommendation, in: Proceedings of the sixteenth ACM international conference on web search and data mining, 2023, pp. 195–203.
[31] X. Li, N. Wang, J. Zeng, Y. Zhong, Z. Shen, Knowledge-enhanced dynamic modeling framework for multi-behavior recommendation, in: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, 2024, pp. 3882–3886.
[32] S. Zhang, H. Chu, J. Li, Y. Zhou, S. Wang, Q. Sun, Dembr: denoising model with memory pruning and semantic guidance for multi-behavior recommendation, in: Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining, 2025, pp. 521–529.
[33] C. Zhai, C. Meng, Y. Yang, K. Zhang, X. Zhao, X. Li, Combinatorial optimization perspective based framework for multi-behavior recommendation, arXiv preprint arXiv:2502.02232, (2025).
[34] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: AISTATS, 2010, pp. 249–256.
[35] H. Fan, F. Zhang, Y. Wei, Z. Li, C. Zou, Y. Gao, Q. Dai, Heterogeneous hypergraph variational autoencoder for link prediction, IEEE Trans. Pattern Anal. Mach Intell. 44 (8) (2022) 4125–4138.
[36] L. Xia, C. Huang, Y. Xu, J. Zhao, D. Yin, J. Huang, Hypergraph contrastive collaborative filtering, in: SIGIR, 2022, pp. 70–79.
[37] N. Wang, D. Liu, J. Zeng, L. Mu, J. Li, HGRec: group recommendation with hypergraph convolutional networks, IEEE Trans. Comput. Soc. Syst. 11 (3) (2024) 4214–4225.
[38] H. Tang, J. Liu, M. Zhao, X. Gong, Progressive layered extraction (PLE): a novel multi-task learning (MTL) model for personalized recommendations, in: Proceedings of the 14th ACM Conference on Recommender Systems, 2020, pp. 269–278.
[39] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: bayesian personalized ranking from implicit feedback, in: UAI, 2009, pp. 452–461.
[40] G. Yuan, F. Yuan, Y. Li, B. Kong, S. Li, L. Chen, M. Yang, C. Yu, B. Hu, Z. Li, et al, TenRec: a large-scale multipurpose benchmark dataset for recommender systems, Adv. Neural Inf. Process. Syst. 35 (2022) 11480–11493.
[41] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T. Chua, Neural collaborative filtering, in: WWW, 2017, pp. 173–182.
[42] M.S. Schlichtkrull, T.N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: ESWC, 2018, pp. 593–607.
[43] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: ICLR, 2015.
[44] C. Gao, X. He, D. Gan, X. Chen, F. Feng, Y. Li, T. Chua, L. Yao, Y. Song, D. Jin, Learning to recommend with multiple cascading behaviors, IEEE Trans. Knowl. Data Eng. 33 (6) (2021) 2588–2601.

Author biography

Yabo Yin is a Master Candidate at College of Computer Science and Engineering, Chongqing University of Technology. He received the B.S. degree in Network Engineering from the Lanzhou Institute of Technology in 2020. His main research interest focuses on machine learning, text mining and recommendation.

Prof. Dr. Xiaofei Zhu is a full professor at College of Computer Science and Engineering, Chongqing University of Technology. He received his PhD degree at the Institute of Computing Technology, Chinese Academy of Science (ICT-CAS) in 2012. Then he spent four years as a Postdoctoral Research Fellow at the L3S Research Center, Leibniz University Hannover. His research interests include web search, data mining and machine learning, and he has published more than 30 papers in international conferences and journals, including the top conferences like SIGIR, WWW, CIKM, TKDE, etc. He has won the Best Paper Awards of CIKM (2011). He serves as area chair, program committees and editorial board of numerous international conferences and journals, including SIGIR, AAAI, CIKM, etc.



Prof. Dr. Jiafeng Guo is currently a Professor in Institute of Computing Technology, Chinese Academy of Sciences, and University of Chinese Academy of Sciences. He received his Ph.D. in Computer Software and Theory from the University of Chinese Academy of Sciences, Beijing, China, in 2009. He has worked on a number of topics related to web search and data mining, including query representation and understanding, learning to rank, and text modeling. His current research focuses on representation learning and neural models for information retrieval and filtering. He has published more than 80 papers in several top conferences/journals such as SIGIR, WWW, CIKM, IJCAI, and TKDE. His work on information retrieval has received the Best Paper Award in ACM CIKM (2011), Best Student Paper Award in ACM SIGIR (2012) and Best Full Paper Runner-up Award in ACM CIKM (2017). Moreover, he has served as the PC member for the prestigious conferences including SIGIR, WWW, KDD, WSDM, and ACL, and the associate editor of TOIS.