Complex & Intelligent Systems
https://doi.org/10.1007/s40747-024-01423-1

ORIGINAL ARTICLE

Multi-behavior collaborative contrastive learning for sequential recommendation

Yuzhe Chen[1] · Qiong Cao[1] · Xianying Huang[1] · Shihao Zou[1]

Received: 12 September 2023 / Accepted: 12 March 2024

Abstract

Sequential recommendation (SR) predicts the user's future preferences based on the sequence of interactions. Recently, some methods for SR have utilized contrastive learning to incorporate self-supervised signals into SR to alleviate the data sparsity problem. Despite these achievements, they overlook the fact that users' multi-behavior interactions in real-world scenarios (e.g., page view, favorite, add to cart, and purchase). Moreover, they disregard the temporal dependencies in users' preferences and their influence on attribute information, leading to models that struggle to accurately capture users' personalized preferences. Therefore, we propose a multi-behavior collaborative contrastive learning for sequential recommendation model. First, we introduce both user-side and item-side attribute information and design an attribute-weight-enhanced attention in multi-behavior interaction scenarios. It enhances the model's ability to capture user's multi-behavior preferences while considering the influence of attribute information. Second, in order to capture users' fine-grained temporal preferences. We divide the interaction sequences into different time scales based on the users' multi-behavior interaction timestamps. In addition, introduce temporal aware attention to generate temporal embeddings for different time scales and effectively integrate them with the user's multi-behavior embeddings. Finally, we design collaborative contrastive learning, which collaboratively captures users' multi-behavior personalized preferences from both temporal and attribute perspectives. This approach alleviates the issue of data sparsity. We conduct extensive experiments on two datasets to validate the effectiveness and superiority of our model.

Keywords Sequential recommendation · Multi-behavior recommendation · Contrastive learning · Graph neural network

Introduction

Recommendation systems aim to address users' personalized preferences and alleviate information overload in many online services such as e-commerce platform, video platform, and music platform [1–3]. Traditional recommendation systems mainly utilize collaborative filtering methods to capture user–item interactions [4, 5]. However, they frequently neglect the sequential information within these interactions [6, 7]. Consequently, researchers are increasingly focusing on sequential recommendations [8].

In recent years, numerous sequential recommendations have emerged with the aim of enhancing the model's ability to capture user preferences [9]. Initially, efforts were made to understand users' overall preferences by analyzing their intricate interaction sequences and subsequently encoding user embedding vectors [10]. However, driven by the rapid advancements in neural network, recent research has utilized neural network to enhance the performance of sequential recommendation [11, 12]. For instance, GRU4Rec [13] utilizes a gate recurrent unit module to encode preference signals from users' interaction sequences. MGNN [14] learns sequential relationships between items using a graph neural network and then employs GRU to assign weights to different behaviors.

Despite their effectiveness, challenges persist in sequential recommendations. First, most existing works have pri-

✉ Qiong Cao
jing5589@163.com

Yuzhe Chen
chenyuzhe5233@163.com

Xianying Huang
wldsj_cqut@163.com

Shihao Zou
z_sh9904@163.com

[1] College of Computer Science and Engineering, Chongqing University of Technology, Hongguang Avenue, Banan District, Chongqing 401320, China

Published online: 15 April 2024

marily focused on encoding sequential behavior patterns for a single behavior. However, they frequently overlook the multi-behavior interactions of users in real-life scenarios, including page view, favorite, and add to the cart. Considering only a single behavior hinders the model from effectively reflecting a user's true preferences. Users' multi-behavior interactions provide a more comprehensive understanding of user preferences.

Furthermore, despite some studies integrating users' multi-behavior into sequential recommendations [15], these studies have primarily neglected the temporal dependencies in users' preferences and their influence on attribute information. First, user preferences are in a constant state of evolution. User preferences vary at each moment, and relying solely on timestamp-based time feature methods [16] cannot capture the fine-grained nature of users' preferences. In Fig. 1, user u_2 engages in sequential behaviors, including viewing dresses, high heels, and favoriting a handbag at time t_1 (initial temporal behaviors). At time t_2, the user's viewing a purple dress and adding the blue dress to the shopping cart collectively influence the purchase behavior at time t_3 (final temporal behavior). Simultaneously, both users and items possess abundant attribute information. Due to variations in users' gender and age, as well as differences in the appearance of item brands, diverse attribute dependencies exist between users and items. For instance, among clothing items, user u_1 prefers jackets, while user u_2 prefers dresses. Attribute information can provide a more precise reflection of users' preferences and intentions.

Moreover, despite auxiliary behaviors aiding in enriching users' preferences for the target behavior (purchase), it still faces challenges related to data sparsity. Contrastive learning, which effectively mitigates data sparsity issues by comparing positive and negative samples from different perspectives, has been widely employed in recent research [17–19]. However, these approaches ignore the user's final temporal behavior, which is influenced by both temporal and attribute information. Simultaneously, the semantics of auxiliary behaviors confuse the semantics of target behaviors during information interactions in contrastive learning. In addition, the model's recommendations lack fairness.

Hence, to address the above issues, we propose the multi-behavior collaborative contrastive learning for sequential recommendation model (MBCCL). First, we introduce user-side and item-side attribute information into the model, leading to the design of the attribute-enhanced multi-behavior embedding encoding module. We propose attribute-weight-enhanced attention (AWE) to generate the weights for the current attributes at each layer of the light graph convolutional neural network, capturing user's personalized preferences influenced by different attribute information. Simultaneously, we partition the interaction sequences into different time scales based on the timestamps of users' multi-behavior interactions. We design the temporal aware attention (TAA) to users' fine-grained temporal preferences. By presetting these temporal embeddings to the multi head attention layer of the Transformer [20], we facilitate efficient fusion between temporal embedding and high-order multi-behavior embedding, thereby capturing users' dynamic preferences across various time scales. Finally, we design collaborative contrastive learning (CCL), which collaboratively captures users' multi-behavior personalized preferences from both temporal and attribute perspectives. This approach alleviates the issue of data sparsity. Our main contributions are as follows:

- We focus on capturing users' personalized preferences by incorporating user-side and item-side attribute information into the model, leading to the design of the attribute-weighted enhanced attention. In addition, we

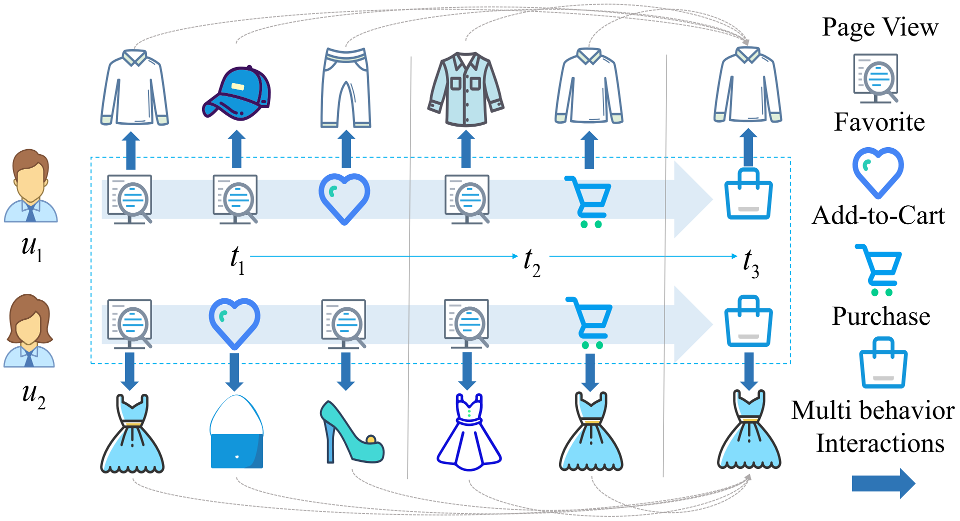


Fig. 1 Example of recommendation. (User's final target behavior (purchase) is influenced by the previous multi-behavior sequential interactions)

introduce the temporal aware encoding module based on the user interaction timestamps to capture users' dynamic personalized preferences.

- We propose the MBCCL model that incorporates multi-behavior into sequential recommendation as auxiliary information, enriching users' preferences from various perspectives. Building upon the encoding of multi-behavior, we design a collaborative contrastive learning approach that collaboratively captures users' personalized preferences from both temporal and attribute perspectives, alleviating data sparsity and improving the fairness of models' recommendation.
- We conduct extensive experiments on two datasets Tmall and Fliggy Trip. Compared to other SOTA baselines, the results demonstrate the effectiveness and superiority of our proposed method.

Related work

Multi-behavior recommendation focuses on the user's multi-behavior interactions in real-world scenarios, while sequential recommendation further explores the sequential interaction patterns between these behaviors to understand the user's interest evolution process more comprehensively. Therefore, we aim to capture users' complex personalized preferences more accurately and comprehensively based on sequential recommendation and using users' multi-behavior interactions. Simultaneously, contrastive learning, as a learning paradigm to address data sparsity, learns high-quality information representation by comparing positive and negative sample pairs. It has found widespread application in recent sequential recommendations. In this section, we summarize the relevant research work from the following research lines: i) sequential recommendation; ii) multi-behavior recommendation; iii) contrastive learning.

Sequential recommendation

Sequential recommendation aims to predict user preferences by leveraging the user interaction sequence [15]. Early research employed Markov chains to predict which items users are likely to interact with next [4]. In recent years, significant efforts have utilized deep learning techniques to capture users' dynamic preferences from their sequential interactions [20]. For example, MGNM [21] utilizes capsule networks and BiLstm to capture users' temporal multi-level interests, while MAE4Rec [22] leverages Transformer to predict user preferences and employs a masked autoencoder to eliminate redundant interactions. Furthermore, the graph neural network captures the user's personalized preferences by encoding user embedding-item embedding with high-order connectivity in the user–item interaction graph, making it widely applicable in sequential recommendation. GES [23] provides a general embedding smoothing framework for sequential recommendation models. LSTCSR [24] constructs an item-item graph with the interaction sequences of different users to obtain information on the collaboration between items in different sequences.

However, the majority of these methods focus solely on a single type of user interaction behavior, neglecting users' multi-behavior interactions in real-life scenarios, such as page view, favorite, and add to the cart. In MBCCL, we employ users' multi-behavior interaction data to capture users' personalized preferences through sequential recommendation.

Multi-behavior recommendation

Multi-behavior recommendation systems leverage users' multi-behavior interaction data to capture personalized preferences [25]. For example, MBN [26] develops a novel multi-behavior network model that captures item correlations and acquires meta-knowledge from multi-behavior basket sequences. MBHT [27] designs a multi-behavior hypergraph-enhanced Transformer framework to capture both short-term and long-term cross-type behavior dependencies. MB-STR [28] uses the Transformer to encode the user's multi-behavior interaction and encode the user's multi-behavior heterogeneous dependencies.

Nevertheless, these methods overlook the fact that users' multi-behavior preferences change over time and are also influenced by attribute information in real-life scenarios. Therefore, we design the temporal aware encoding module and the attribute-enhanced encoding module to generate dynamic user preferences that incorporate attribute information.

Contrastive learning

Contrastive learning learns information representation by comparing pairs of positive and negative samples to generate high-quality embedding while alleviating the problem of data sparsity. It has been widely used in visual representation learning [29] and natural language processing [30]. Recently, some studies have proposed introducing contrastive learning into a recommendation system [31]. For example, CML [17] pioneered the integration of contrastive learning into multi-behavior recommendation systems. MCLSR [32] proposed a multi-level comparison learning framework to effectively learn self-supervised signals. It constructs three views in addition to the sequence view and utilizes a multi-level comparison mechanism to learn collaborative information. EC4SREC [33] uses the interpretation method to determine the importance of the commodity in the user sequence and derive the positive and negative sequences accordingly.

Following this research line, we propose collaborative contrastive learning (CCL), which collaboratively captures users' personalized preferences from both temporal and attribute perspectives.

Preliminary

In our sequential recommendation scenario, suppose we have the users set $U = \{u_1, \ldots, u_m\}$ and the items set $I = \{i_1, \ldots, i_n\}$. Introduce and define the set of behaviors B. We consider the behavior to be predicted (purchase) as the target behavior b, while other behaviors (such as page views, favorites, and adding to the cart) are considered auxiliary behaviors b'. The user's multi-behavior interaction sequences are $\{Y_u^b, \ldots, Y_u^B\}$. MBCCL leverages temporal and attribute information to enhance the capture of users' fine-grained preferences. We further define the following.

Multi-behavior temporal user–item interaction graph By incorporating temporal information that reflects users' evolving preferences over time, we transform the multi-behavior user–item interaction data based on their interaction timestamps $t_{u,i,b}$ into a graph structure, which we define as $G_b^t = (V, E_b)$. Where V represents the set of nodes, including users and items, and E_b represents the set of edges, including users' multi-behavior interactions. In Fig. 2, dashed lines represent the initial interactions between users and items at time t_1 while solid lines represent the final interactions at time t_3.

Multi-behavior attribute user–item interaction graph In real-world scenarios, users' multi-behavior interactions are influenced by attribute information. To encode fine-grained user preferences, we transform user interaction data along with user-side and item-side attribute information into a graph structure, defined as $G_b^a = (V, E_b)$, where V represents the set of nodes, including users, items, and attribute information, and E_b represents the set of edges representing the multi-behavior of users. Each user node is connected to the item nodes and attribute nodes it interacts with, and the same applies to the item side.

Our research question can be formalized as follows:

Input Sequences of multi-behavior interactions $\{Y_u^b, \ldots, Y_u^B\}$.

Output The learning function estimates the probability that user u_m will interact with item i_n with the target behavior at the future $t + 1$ time step.

Our approach

The structure of our proposed MBCCL model is shown in Fig. 3, which consists of four key modules: (1) initial embedding layer; (2) temporal aware multi-behavior embedding encoding; (3) attribute-enhanced multi-behavior embedding encoding; and (4) collaborative contrastive learning.

Initial embedding layer

In order to capture the user's preferences, we utilize the initial embedding layer to generate low-dimensional embedding for users E_u and items E_i:

$$E_u = \text{Embedding}(\text{ID}_{\text{user}});\ E_i = \text{Embedding}(\text{ID}_{\text{item}}) \qquad (1)$$

where Embedding indicates to map the user's and item's ID to a low-dimensional embedding space for generating user embedding E_u and item embedding E_i, based on the embedding dimension d and batch num n. We use the Xavier initializer [34] to generate initial parameters for the embedding matrix. In the training process of MBCCL, the parameters of the initial embedding layer will be learned to better capture the user's preferences and behavior patterns. In

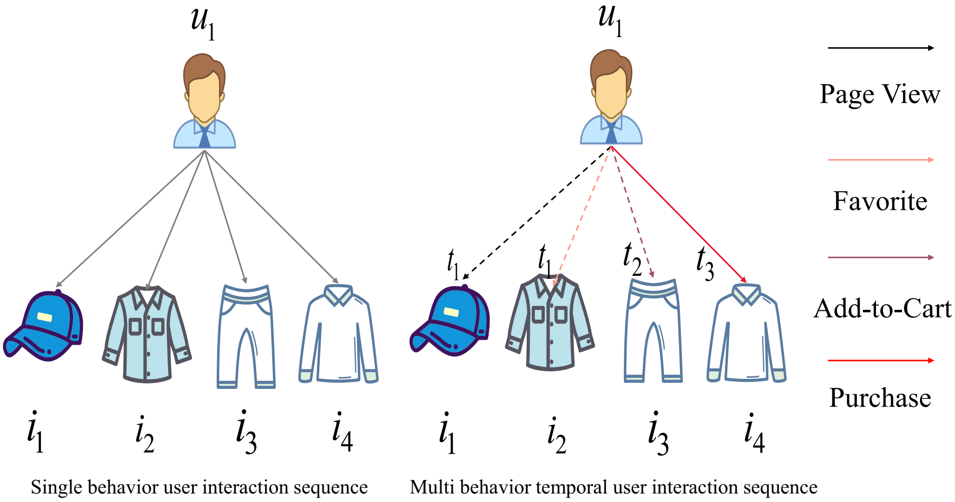


Fig. 2 Example of the user's single-behavior interaction sequence and multi-behavior temporal interaction sequence integrating attribute information

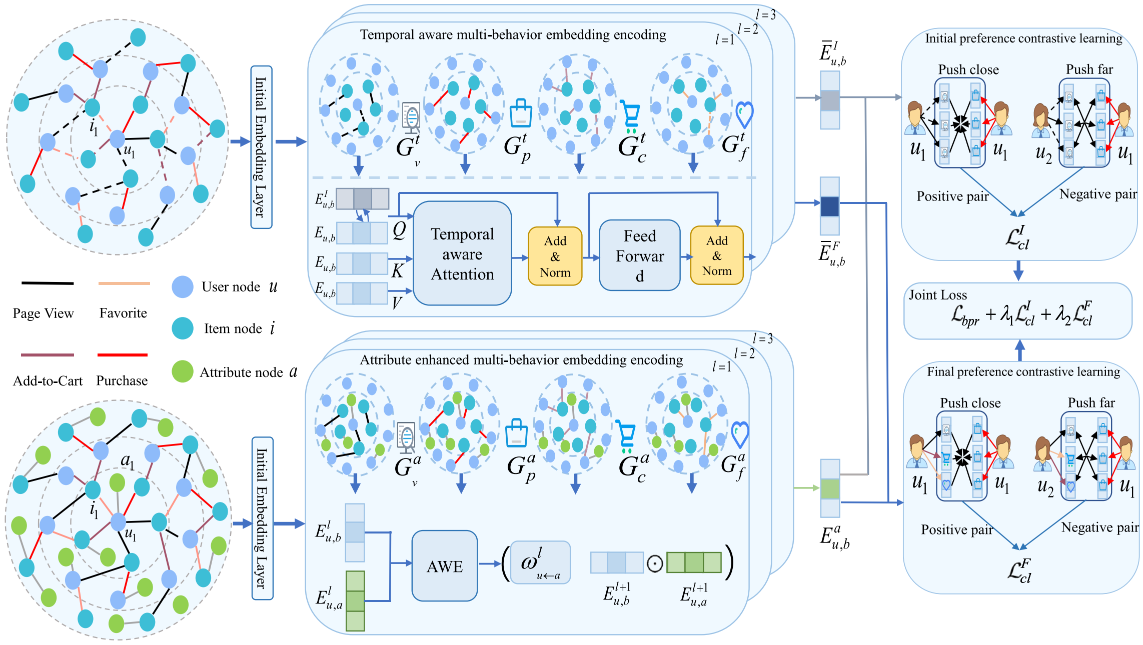


Fig. 3 The overall architecture of MBCCL. (i) Temporal aware multi-behavior embedding encoding, which generates the initial temporal and final temporal multi-behavior embedding; (ii) attribute-enhanced multi-behavior embedding encoding, which generates the attribute weight to enhance multi-behavior embedding; (iii) collaborative contrastive learning, which collaboratively captures users' multi-behavior personalized preferences from both temporal and attribute perspectives

addition, a fully connected layer is utilized to generate initial attribute embedding for users and items [35]. Specifically, the user's initial attribute embedding $E_{u,a}$ is calculated as follows:

$$E_{u,a} = f\left(WX_z + b\right) \tag{2}$$

where X indicates the one-hot encoding of the user's attribute information, z is the number of attributes of the current user, W and b are parameters of the fully connected layer, and f is the activation function. As the user has multiple attributes (such as age and gender), we average the attribute embedding to generate the user's initial attribute embedding. The same operation applies to the item side.

Temporal aware multi-behavior embedding encoding

In multi-behavior interaction scenarios, users' personalized preferences change over time, and their ultimate target behavior (purchase) is influenced by their previous sequential interactions with multi-behavior. In addition, the weights between these different behaviors are not the same. For example, in Fig. 1, the "add to cart" behavior at time t_2 is more important than the "page view" behavior at time t_1. To capture users' dynamic preferences and encode the importance of different behaviors, focus on the multi-behavior temporal user–item interaction graph. We design a temporal aware multi-behavior embedding encoding module based on the light graph convolutional neural network (LGCN) [36].

Higher order multi-behavior embedding encoding

In the context of multi-behavior interaction, there are complex relationships between user behaviors, and each behavior has its own unique characteristics. For example, on e-commerce platforms, the page view behavior is predominant, and the purchase behavior is uncommon. To concentrate on each behavior, we iteratively aggregate information on four multi-behavior temporal interactions graphs $(G^t_v, G^t_f, G^t_c, G^t_p)$ to generate multi-behavior embedding representations for users and items. This enables us to learn high order dependency relationships within user interactions. Users' and items' multi-behavior embedding generated at the l layer are computed as follows:

$$E^{l+1}_{u,b} = \sum_{i\in N_{u,b}} \phi E^{l+1}_{i,b} \tag{3}$$

where l indicates the number of LGCN layers, $E^{l+1}_{u,b}$ is multi-behavior user embedding with higher order connectivity at the l layer, when $l = 0$, the embeddings are the initial user and item embedding generated in Sect. 4.1, and $N_{u,b}$ are the neighboring nodes for users under the current behavior. $\phi = 1/\sqrt{N_{u,b}N_{i,b}}$ is the symmetric normalization term to avoid that the size of the embedding increases with the number of layers of the graph convolution and to improve the stability of the model. The same operation is applied to the item side.

At the same time, in multi-behavior scenarios, different behaviors have their own distinct impacts on predicting user preferences. Using an e-commerce platform as an example, the purchase behavior often more accurately reflects a user's actual preferences compared to the page view behavior. To more precisely leverage various behaviors in capturing user preferences, we propose an adaptive weight generation method that enables the model to automatically learn the importance of different behaviors in prediction. The calculation of behavior weights is shown as follows:

$$\alpha_{u,b} = \frac{w_b \cdot n_{u,b}}{\sum_{s\in N_B} w_s \cdot n_{u,s}};\ E^{l+1}_{u,b} = \alpha_{u,b} \cdot E^{l+1}_{u,b} \tag{4}$$

where w_b indicates user weight under behavior b, which is the same for all users. $n_{u,b}$ indicates the number of times the user interacted with the item under the behavior b and $n_{u,s}$ indicates the total number of user interactions. Subsequently, we aggregate the multi-behavior embedding of users and items to generate high-level connectivity embedding:

$$E^{l+1}_{u,b} = \sigma\left(W_g \cdot E^{l+1}_{u,b}\right) \tag{5}$$

where σ indicates the activation function PReLu, W_g indicates the parameter matrix, and the same operation is applied to the item side. The computation of the final multi-behavior embeddings for users and items is shown as follows:

$$E_{u,b} = \frac{1}{l}\sum_{l=1}^{L} E_{i,b};\ E_{i,b} = \frac{1}{l}\sum_{l=1}^{L} E_{u,b} \tag{6}$$

Temporal aware attention (TAA)

Users' final purchase preferences may deviate from their initial preferences. For example, in Fig. 1, the user's preferences at t_1 include tops, hats, and pants. In contrast, before the purchase, the user's preference is only for tops. Unlike CLSR [37], which divides user preferences into short-term and global preferences, given that user preferences change over time, we contend that global preferences cannot effectively reflect a user's personalized dynamic preferences. We divide the user's multi-behavior interactions into initial interactions $t^I_{u,i,b}$ and final interactions $t^F_{u,i,b}$ based on the timestamps $t_{u,i,b}$. Design a temporal aware attention and use the slot mapping function $\tau(\cdot)$ to generate two temporal embeddings for users, as shown in Fig. 4. The calculation is as follows:

$$T^I_{u,i,b} = \sin\left(\tau\left(t^I_{u,i,b}\right)\cdot \bar{w}_1\right)$$
$$T^F_{u,i,b} = \cos\left(\tau\left(t^F_{u,i,b}\right)\cdot \bar{w}_2\right)$$

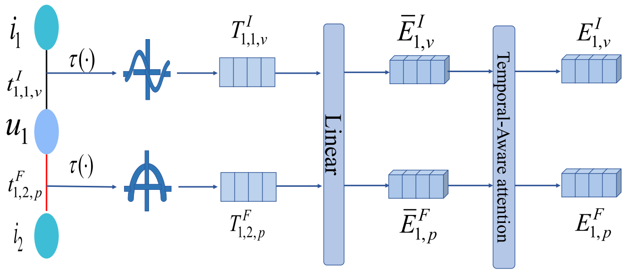


Fig. 4 Example of temporal aware attention

$$\bar{w}_1 = \frac{1}{10,000^{\frac{2l}{d_t}}};\ \bar{w}_2 = \frac{1}{10,000^{\frac{2l+1}{d_t}}} \tag{7}$$

where w_1 and w_2 represent the parameter matrix, and d_t represents the temporal embedding dimension. Subsequently, we utilize a linear layer to align the acquired initial temporal embedding and final temporal embedding with the high-order multi-behavior user embedding:

$$\bar{E}^I_{u,b} = \text{Linear}\left(T^I_{u,i,b}\right);\ \bar{E}^F_{u,b} = \text{Linear}\left(T^F_{u,i,b}\right) \tag{8}$$

We put the multi-behavior user embeddings generated in Sect. 4.2.1 into query/key/value vectors as $Q = E_{u,b}W_Q$, $K = E_{u,b}W_k$, $V = E_{u,b}W_V$. Then, we preset the initial temporal embedding sequence $\bar{E}^I_{u,b} = \{\bar{E}^I_{u_1,b}, \ldots, \bar{E}^I_{u_m,b}\}$ and the final temporal embedding sequence $\bar{E}^F_{u,b} = \{\bar{E}^F_{u_1,b}, \ldots, \bar{E}^F_{u_m,b}\}$ at the multi-head attention layer of the Transformer [20]. This method is referred to as temporal aware attention (TAA) and is implemented as shown in the following equation:

$$\begin{aligned} Y_I &= \text{TAA}\left([E_{u,b}\|\bar{E}^I_{u,b}], E_{u,b}, E_{u,b}\right) \\ &= \text{softmax}\left(\frac{[\bar{E}^I_{u,b}; Q]K}{\sqrt{d}}\right)V \end{aligned} \tag{9}$$

where Y_I is the result of the weights of a layer of initial tempora-aware attention, and $\|$ denotes concat operation. In addition, the same operation applies to the final tempora-aware attention. In this way, multi-head attention is combined to obtain the output of the multi-head attention layer:

$$\text{Multihead }(Y_I) = \text{Concat}\left(Y^1_I, \ldots, Y^h_I\right)W \tag{10}$$

where $\{Y^1_I, \ldots, Y^h_I\}$ are the output of each attention layer, h are the layers of attention. We normalize the output results of the multi-head attention layers using a residual connection layer with regularization. Then, we use a feed-forward network to obtain the final attention output:

$$\begin{aligned} N &= \text{Norm}\left(Y_I + \text{ Multihead }(Y_I)\right) \\ F &= \max\left(0, NW_1 + b_1\right)W_2 + b_2 \end{aligned} \tag{11}$$

where W is the weight parameter and b is the bias. We use temporal aware attention to effectively fuse the high-order user–item multi-behavior embeddings with the two types of temporal embeddings, capturing the user's evolving personalized preferences over time. The final computed initial temporal user multi-behavior embeddings $E^{I}_{u,b}$ and final temporal user multi-behavior embeddings $E^{F}_{u,b}$ are calculated as follows:

$$\begin{aligned} E^{I}_{u,b} &= \mathrm{TAA}\left(E_{u,b} \leftarrow \bar{E}^{I}_{u,b}\right) \\ E^{F}_{u,b} &= \mathrm{TAA}\left(E_{u,b} \leftarrow \bar{E}^{F}_{u,b}\right) \end{aligned} \tag{12}$$

Attribute-enhanced multi-behavior embedding encoding

On online platforms, there is rich attribute information associated with user–item interactions. For instance, user attributes such as age and gender and item attributes such as brand and appearance play a significant role in shaping user preferences. Users with different attributes may exhibit different preferences for the same item. Instead of solely generating attribute embeddings as done in AGCN [35], we propose an attribute-enhanced multi-behavior embedding encoding module to enhance user–item embeddings and capture users' fine-grained preferences.

Attribute-weight-enhanced attention (AWE)

Different attribute information holds varying importance for users. To better utilize diverse attribute information in capturing fine-grained user preferences and providing more personalized recommendations, we design the attribute-weighted enhancement attention (AWE) based on LGCN. Aim to generate fused attribute-weighted user and item embeddings based on the different attribute information:

$$\begin{aligned} \omega^{l}_{u\leftarrow a} &= \mathrm{softmax}(\sigma(W^{l}(E^{l}_{u,b} \| E^{l}_{u,a} + b^{l}))) \\ \omega^{l}_{i\leftarrow a} &= \mathrm{softmax}(\sigma(W^{l}(E^{l}_{i,b} \| E^{l}_{i,a} + b^{l}))) \end{aligned} \tag{13}$$

where $\omega^{l}_{u\leftarrow a}$ is the attribute weight at the l layer. σ is the activation function, and W^{l}, b^{l} are the weight matrix and bias.

Weight-enhanced embedding aggregation

In the multi-behavior attribute user–item interaction graph, each user node is connected to the item nodes it interacts with, as well as the attribute nodes. Simultaneously, item nodes are connected to their corresponding attribute nodes and the user nodes they interact with. We utilize LGCN to process the four multi-behavior attribute user–item interaction graphs ($G^{a}_{v}, G^{a}_{f}, G^{a}_{c}, G^{a}_{p}$). This is done to generate high-order embeddings for users, items, and attribute. The computation process is as follows:

$$\begin{aligned} E^{(l+1)}_{u,a} &= \tfrac{1}{N_{u,a}} \sum_{u\in N_{u,a}} E^{(l)}_{u,b} \\ E^{(l+1)}_{u\leftarrow a,b} &= \sum_{a\in N_{u,a}} \omega^{l}_{u\leftarrow a} E^{(l)}_{u,b} \odot E^{(l)}_{u,a} \end{aligned} \tag{14}$$

where $E^{(l+1)}_{u,a}$ is the attribute embedding, $E^{(l+1)}_{u\leftarrow a,b}$ is the multi-behavior user embedding enhanced by attribute weight, l represents the number of LGCN layers, when $l = 0$, the embeddings are the initial user–item and attribute embedding which generated in Sect. 4.1, $N_{u,a}$ represents neighboring attribute nodes for users, and similar operations are applied to the item side. After multi-layers of graph convolution operations, the user and item embeddings enhanced by attribute weights are calculated as follows:

$$E^{a}_{u,b} = \tfrac{1}{l} \sum_{l=1}^{L} E^{(l+1)}_{u\leftarrow a,b};\ E^{a}_{i,b} = \tfrac{1}{l} \sum_{l=1}^{L} E_{i\leftarrow a,b} \tag{15}$$

Collaborative contrastive learning

Contrastive learning, as a learning paradigm addressing data sparsity issue learns high-quality information representation by comparing positive and negative sample pairs. It has found widespread application in recent sequential recommendations [38]. Nonetheless, these approaches tend to overlook users' personalized preferences, resulting in a diminished diversity in model recommendations. For instance, CML [17], the pioneering use of contrastive learning in multi-behavior recommendation, considered target behavior embedding and auxiliary behavior embedding as positive and negative samples. However, they neglect the dynamic preferences of users. CLSR [37], employing multi-view contrastive learning from graphs and sequential structures, does take temporal preferences into account. However, it fails to capture the implicit information within a user's multi-behavior interactions, influenced by rich attribute information. To address this, we introduce collaborative contrastive learning (CCL) to collaboratively capture users' personalized preferences from both temporal and attribute perspectives. CCL models the evolution of user interests, addresses data sparsity issues and enhances the diversity and fairness of model recommendations.

User-preference aggregation

In the encoding processes of attribute-enhanced multi-behavior embedding and temporal aware multi-behavior embedding, we acquire attribute-enhanced multi-behavior user and item embedding sequences, initial temporal user embedding sequences, and final temporal user embedding sequences. We aggregate them to generate fused attribute-weighted initial temporal preferences and final temporal preferences:

$$\tilde{E}^{I}_{u,b} = E^{a}_{u,b} \| E^{I}_{u,b};\ \tilde{E}^{F}_{u,b} = E^{a}_{u,b} \| E^{F}_{u,b} \tag{16}$$

Initial preference contrastive learning

Through the introduction of auxiliary behaviors and facilitating information interaction with the target behavior, we efficiently address the data sparsity issue related to the target behavior. During the initial stages of user interactions, user preferences showcase diversity, and various behaviors carry different levels of significance for users. Hence, in the context of initial preference contrastive learning, we define positive samples as embeddings of the target behavior and multi-auxiliary behavior embeddings for the same user $\{\tilde{E}^{I}_{u_1,b}, \tilde{E}^{I}_{u_1,v}/\tilde{E}^{I}_{u_1,f}/\tilde{E}^{I}_{u_1,c}\}$. Simultaneously, within a batch, we randomly select users and define negative samples as embeddings of the target behavior and multi-auxiliary behaviors for different users $\{\tilde{E}^{I}_{u_1,b}, \tilde{E}^{I}_{u_2,v}\}$. We define the user's initial preference contrastive loss $\mathcal{L}^{I}_{cl,u}$ based on InfoNCE [39] as follows:

$$\mathcal{L}^{I}_{cl,u} = \sum_{u \in U} -\log \frac{\exp\left(\text{cosine}\left(\tilde{E}^{I}_{u_1,b}, \tilde{E}^{I}_{u_1,v}\right)/\tau\right)}{\sum_{u \in U} \exp\left(\text{cosine}\left(\tilde{E}^{I}_{u_1,b}, \tilde{E}^{I}_{u_2,v}\right)/\tau\right)} \tag{17}$$

where τ represents the temperature hyperparameter used to adjust the scale of the softmax function, with the same operation applied to the item side. When analogously combined with the contrastive loss of the item side, the initial preference contrastive loss is $\mathcal{L}^{I}_{cl} = \mathcal{L}^{I}_{cl,u} + \mathcal{L}^{I}_{cl,i}$. The initial preference contrastive learning loss is composed of the sum of each pair of auxiliary behaviors (page view, favorite, add to cart) with the target behavior (purchase).

Final preference contrastive learning

The user's initial temporal preference and final temporal preference are not the same in the end. Preferences at different times have different importance for model prediction. Meanwhile, during the initial preference contrastive learning process, the model might capture too much semantic information from auxiliary behaviors (e.g., page view, collecting, and adding to cart), thereby confusing the semantic information of the target behavior (purchase) itself. To mitigate the semantic influence of auxiliary behaviors, we concatenate auxiliary embeddings $\tilde{E}^{F}_{u,b'} = \tilde{E}^{F}_{u,v} \| \tilde{E}^{F}_{u,f} \| \tilde{E}^{F}_{u,c}$ to consolidate the semantic information of the target behavior.

In the final preference contrastive learning, we define positive samples as embeddings of the same user's target behavior and auxiliary behaviors $\{\tilde{E}^{F}_{u_1,b}, \tilde{E}^{F}_{u_1,b'}\}$, and negative samples as embeddings of different users' target behavior and auxiliary behaviors $\{\tilde{E}^{F}_{u_1,b}, \tilde{E}^{F}_{u_2,b'}\}$. The user's final preference contrastive loss is defined as follows $\mathcal{L}^{F}_{cl,u}$:

$$\mathcal{L}^{F}_{cl,u} = \sum_{u \in U} -\log \frac{\exp\left(\text{cosine}\left(\tilde{E}^{F}_{u_1,b}, \tilde{E}^{F}_{u_1,b'}\right)/\tau\right)}{\sum_{u \in U} \exp\left(\text{cosine}\left(\tilde{E}^{F}_{u_1,b}, \tilde{E}^{F}_{u_2,b'}\right)/\tau\right)} \tag{18}$$

Analogously combining the loss of item side, we obtain the final objective function of final preference contrastive loss as $\mathcal{L}^{F}_{cl} = \mathcal{L}^{F}_{cl,u} + \mathcal{L}^{F}_{cl,i}$.

Joint loss

In the learning phase of MBCCL, we utilize the original Bayesian ranking (BPR) loss $\mathcal{L}_{\text{bpr}}$, initial preference contrastive loss$\mathcal{L}^{I}_{\text{cl}}$, and final preference contrastive loss loss$\mathcal{L}^{F}_{\text{cl}}$ to jointly optimize the model for the main supervised task. This ensures that the model minimizes the difference between the positive and negative sample predictions of the users. We define the BPR loss $\mathcal{L}_{\text{bpr}}$ as follows:

$$\mathcal{L}_{\text{bpr}} = \sum_{n \in U} \sum_{i \in \mathbb{N}_u} \sum_{i' \notin \mathbb{N}_u} -\log\left(\sigma\left(y^{b}_{u,i^{+}} - y^{b}_{u,i^{-}}\right)\right) \tag{19}$$

where $y^{b}_{u,i^{+}}, y^{b}_{u,i^{-}}$ are the positive and negative sample of BPR loss. The loss and the prediction of the model are calculated as follows:

$$\mathcal{L} = \mathcal{L}_{\text{bpr}} + \lambda_1 \mathcal{L}^{I}_{cl} + \lambda_2 \mathcal{L}^{F}_{cl} \tag{20}$$

where λ_1, λ_2 indicate the weights of the collaborative contrastive loss, which aim to highlight the importance of different moments of user interaction. We use the Adam optimizer with stochastic gradient descent to train our model during the training process.

Experiments

In this section, we conduct experiments on two datasets to evaluate the performance of our MBCCL by answering the following research questions:

- *RQ1* How does MBCCL perform when competing with various recommendation baselines on different datasets?
- *RQ2* How do the designed three modules in our MBCCL frame work affect the recommendation performance?
- *RQ3* How effective are the multi-behavior and attribute information introduced in our MBCCL framework?

Table 1 Statistics of experimented datasets

Dataset	User	Item	Multi-behavior types	User attribute	Item attribute	Interactions
Tmall	7775	5953	View, fav, cart, buy	Age, gender	Category, brand, seller	60,087
Fliggy trip	17,307	4923	View, cart, buy	Age, occupation, habitual city	Category, city	161,316

- *RQ4* How effective is the collaborative contrastive learning we designed?
- *RQ5* How do different configurations of key hyperparameters impact the performance of MBCCL framework?

Experimental settings

Datasets

We utilize two recommendation datasets collected from real-world scenarios. The Tmall dataset is provided by Tmall stores, and the Fliggy Trip dataset is provided by the Fliggy platform. These datasets contain users' multi-behavior as well as rich attribute information for users and items. Table 1 presents the data distribution for both datasets. In our experiments, users and items with fewer than five interactions were filtered out.

Evaluation metrics

The experiment uses two evaluation metrics widely used in recommendation tasks: normalized discounted cumulative gain (NDCG) and hit ratio (HR). In our experiments, we adopted the leave-one-out evaluation approach, which has been widely used in previous works.

Methods for comparison

To evaluate the performance of our MBCCL, we compare it with various SOTA recommendation baselines from five groups: single-behavior recommendation, multi-behavior recommendation, sequential recommendation, recommendation with contrastive learning and incorporate external knowledge recommendation.

Single-behavior recommendation

- *IMP-GCN* [12] It aggregates users with similar preferences into a user–item interaction graph and uses graph neural networks to encode embeddings for users and items.
- *STAM* [40] It designs a temporal encoding module that incorporates temporal information into neighboring nodes and uses neural networks to generate temporal embeddings.

Multi-behavior recommendation

- *EHCF* [41] It applies a new non-sampling migration learning, i.e., efficient heterogeneous collaborative filtering, to sequence recommendation models.
- *GHCF* [42] It is based on users' multi-behavior interactions and incorporates the weights of behaviors into the graph convolution operation for iterative aggregation.
- *MBGCN* [43] It introduced a heterogeneous graph that combines multi-behavior, allowing user embedding to incorporate weights specific to certain behaviors, facilitating behavior-aware embedding propagation.
- *MBGMN* [44] It customizes the meta-learning paradigm for multi-behavior relationship learning and performs embedding propagation in graph neural networks.

Sequential recommendation

- *SASREC* [1] It introduces a self-attention-based sequential model to capture long-term semantic information.
- *Bert4rec* [45] It predicts user's interest in products by extracting their historical interests from user's past behaviors.
- *SURGE* [46] It uses metric learning to reconstruct loose item sequentials into a dense item-item interest graph, aggregating different types of preferences from long-term user behavior into clusters within the graph.

Recommendation with contrastive learning

- *CML* [17] It pioneered the application of contrastive learning in multi-behavior recommendation models for the first time.
- *CLSR* [37] It separates user preferences into long-term and short-term preferences and decouples them through self-supervised learning.
- *MMCLR* [19] It enhances the multi-behavior recommendation model by introducing a multi-view contrastive learning approach based on both sequence and graph structures.

Incorporating external knowledge recommendation

- *KGAT* [47] It combines user–item interaction graphs with knowledge graphs and uses attention mechanisms to differentiate the importance of different neighbor nodes.

- *AGCN* [35] It incorporates attribute information into the user–item interaction graph and proposes an adaptive graph convolutional network for joint item recommendation and attribute inference.

Parameter settings

Our proposed model is implemented on the Pytorch framework. The hyperparameters are set as follows: the number of multi-head attention heads in TAA is 6. The dropout in both Tmall and Fliggy Trip is 0.7. The learning rate in Tmall is 1e−3, and that in Fliggy Trip is 1e−2. batch size is set to 128 in Tmall and Fliggy Trip. The embedding vector dimension is 32 and the temporal embedding vector dimension is 16. The reports of the models we implemented are all based on the average scores of 5 random runs on the test set.

Performance comparison (RQ1)

We give detailed evaluation results for all methods on different datasets in Table 2, where the results for MBCCL are highlighted in bold. The main observations are as follows:

- The performance gap between MBCCL and single-behavior recommendation models (IMP-GCN, STAM) underscores the effectiveness of incorporating multi-behaviors as auxiliary information to capture users' personalized preferences.
- Compared to multi-behavior recommendation models (EHCF, GHCF, MBGCN, and MBGMN), MBCCL demonstrates superior performance. Existing multi-behavior recommendations focus on utilizing multi-behavior interaction data to capture users' dependencies on multi-behaviors, often neglecting temporality. This validates the effectiveness of our temporal aware encoding module in capturing users' dynamic preferences.
- Among the various comparison methods, it is evident that the injection of multi-behavior information into sequential recommendations significantly enhances performance compared to traditional sequential recommendation models (SASREC, Bert4rec, and SURGE). Traditional sequential recommendation learns the user's sequential interaction pattern based on the interaction sequence. Building upon this, we use multi-behavior information to further enrich the user's sequential interaction pattern. Concurrently, we introduce rich attribute information on the user side and item side to capture users' fine-grained preferences.
- We assess the performance of MBCCL against several representative recommendations with contrastive learning (CML, CLSR, and MMCLR). Nevertheless, prior studies leveraging contrastive learning for performance improvement often overlook the influence of rich attribute information on user preferences. In addition, the weights assigned to user preferences at the initial moment and the final moment differ. Therefore, we propose collaborative contrastive learning (CCL), encompassing both initial preference contrastive learning and final preference contrastive learning. This approach collaboratively captures users' personalized preferences from both temporal and attribute perspectives.

Table 2 Experimental results on Tmall and Fliggy trip datasets

Models	Tmall				Fliggy Trip			
	HR	Imprv	NDCG	Imprv	HR	Imprv	NDCG	Imprv
IMP-GCN	0.041	5.34	0.024	6.21	0.084	1.17	0.059	1.23
STAM	0.049	4.12	0.029	4.86	0.091	1.04	0.065	1.04
EHCF	0.070	2.72	0.033	4.24	0.071	1.61	0.059	1.23
GHCF	0.161	0.62	0.078	1.21	0.104	0.78	0.071	0.85
MBGCN	0.187	0.39	0.084	1.05	0.083	1.24	0.062	0.51
MBGMN	0.196	0.33	0.092	0.88	0.114	0.67	0.076	0.73
SASRec	0.137	0.90	0.056	2.08	0.098	0.89	0.081	0.57
Bert4rec	0.164	0.59	0.059	1.93	0.128	0.45	0.105	0.25
SURGE	0.223	0.17	0.121	0.42	0.131	0.41	0.117	0.13
CML	0.205	0.26	0.096	0.80	0.119	0.56	0.087	0.51
CLSR	0.220	0.18	0.127	0.36	0.125	0.49	0.094	0.40
MMCLR	0.235	0.11	0.136	0.27	0.143	0.41	0.121	0.09
KGAT	0.039	5.63	0.021	6.40	0.105	0.77	0.073	0.81
AGCN	0.131	0.95	0.049	2.23	0.123	0.51	0.090	0.46
$MBCCL$	0.261	–	0.173	–	0.186	–	0.132	–

HR@N, NDCG@$N(N = 20)$

- The performance gap observed between MBCCL and other models (KGAT, AGCN) underscores the advantages of our attribute-weight-enhanced attention approach. Leveraging attribute embedding to generate attribute weights on graph neural networks proves effective in enhancing model performance and fairness.

Model ablation study (RQ2)

To explore the distinct contributions of modules in MBCCL, we conduct an ablation study on two datasets. We examined the following configurations:

- *w/o TAE (temporal aware encoding)* We removed the temporal aware encoding module and utilized the initial embeddings of users and items for iterative aggregation in the graph convolutional neural network.
- *w/o AWE (attribute-enhanced encoding)* We removed the attribute enhanced encoding module.

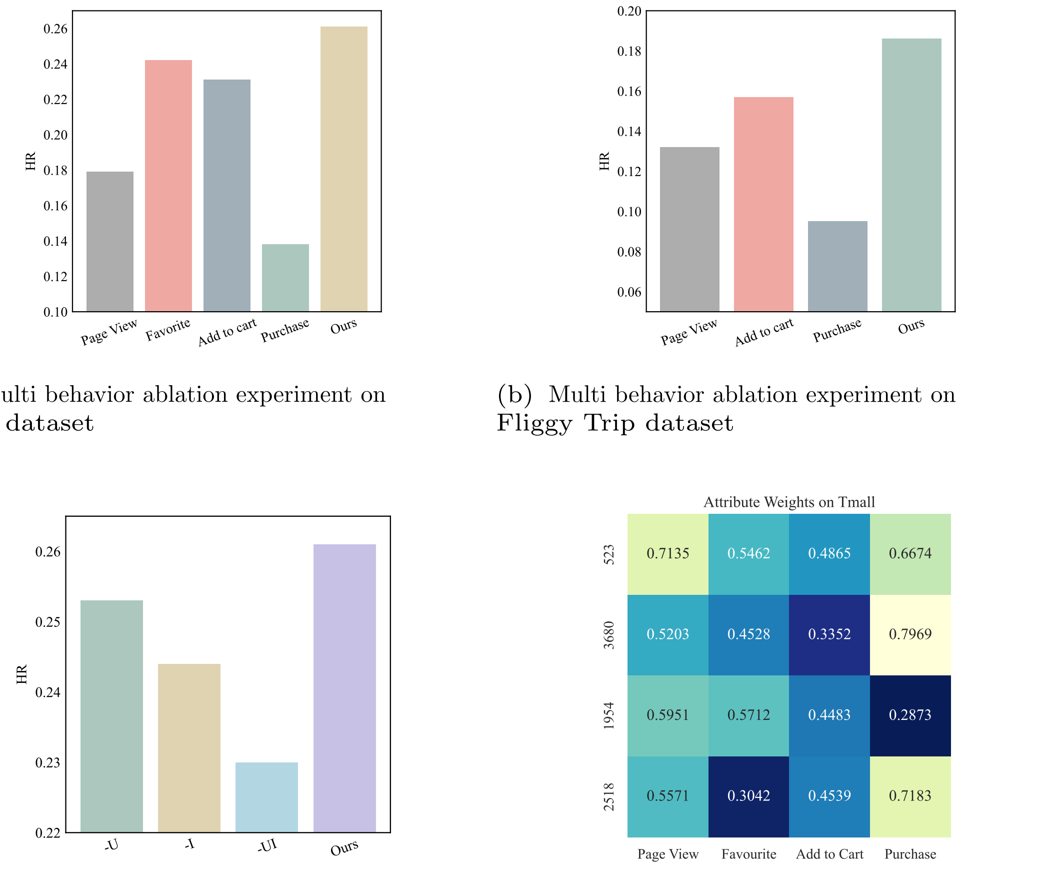
Table 3 Ablation study on two datasets, where HR@N, NDCG@N $(N = 10)$

	Tmall		Fliggy trip	
	HR	NDCG	HR	NDCG
MBCCL	0.261	0.173	0.186	0.132
w/o TAE	0.254	0.169	0.179	0.125
w/o AWE	0.227	0.138	0.159	0.105
w/o CCL	0.235	0.145	0.162	0.113

- *w/o CCL (collaborative contrastive learning)* We removed the collaborative contrastive learning and used only the BPR loss during model training.

Table 3 shows the results of the ablation experiments:

1. By comparing MBCCL with "w/o TAE," we observed that the introduction of temporal aware encoding into the model, along with the use of temporal aware attention, enables the model to capture dynamic user preferences



(a) Multi behavior ablation experiment on Tmall dataset

(b) Multi behavior ablation experiment on Fliggy Trip dataset

(c) Attribute information ablation experiment on Tmall dataset

(d) Attribute weights visualization on Tmall dataset

Fig. 5 Multi-behavior and attribute information impact studies

that evolve over time, thereby improving the accuracy of model predictions.

2. By comparing MBCCL with "w/o AWE," we found that utilizing attribute-weight-enhanced user and item embedding performed better than using only the initial embedding of users and items. User interactions with multi-behaviors are influenced by rich attribute information related to users, such as age and gender, as well as attributes of items such as brand and appearance. By incorporating attribute information, we effectively enhance the model's performance and increase the fairness of the model.
3. By comparing MBCCL with "w/o CCL," we observed that our collaborative contrastive learning, designed from both temporal and attribute perspectives, effectively mitigates the data sparsity issue of target behaviors. In addition, starting from real-world scenarios and dividing user interactions into initial interactions and final interactions based on timestamps, we designed initial preference contrastive learning and final preference contrastive learning to effectively capture the user's fine-grained preferences.

Effect of multi-behavior and attribute information (RQ3)

From Fig. 5a, b, we can observe that the purchase behavior contributes the most to the model's predictions, followed by page view behavior. In real-world scenarios, user's multi-behavior can better reflect their personalized preferences. These results indicate that the impact of behaviors is significant in sequential recommendation. By introducing multi-behaviors, we effectively enrich user preferences from various perspectives and enhance the model's recommendation performance.

We conduct an ablation experiment focusing with attribute information, as illustrated in Fig. 5c. The results unequivocally indicate that item attributes wield the most substantial influence on the model's performance. Attribute information notably enhances the model's overall performance. Diverse attribute information exerts varying impacts on a user's interactions with multi-behaviors. Hence, we introduce attribute-weight-enhanced attention to encode attribute weights.

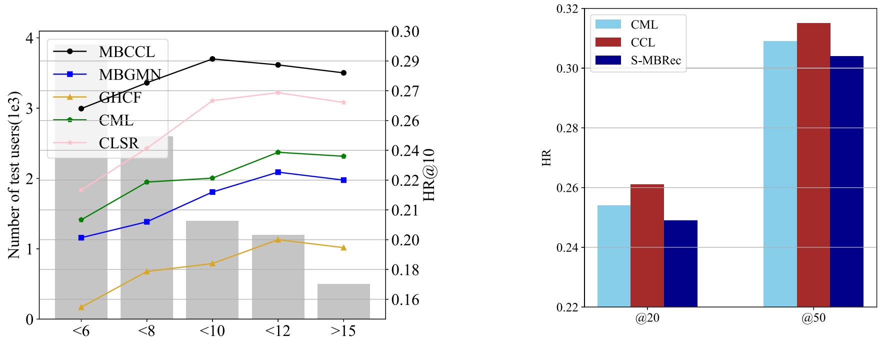
To delve into the user's nuanced preferences, we visually represent attribute weights, as depicted in Fig. 5d. For instance, the attribute weight for a user's purchase behavior on an item with ID 3680 is 0.7969, and the favorite behavior weight is 0.4528, signifying the user's heightened inclination to purchase the item based on its attribute information. The attribute weight enhancement attention serves to capture users' fine-grained preferences and enhance the fairness of model recommendations.

Effect of collaborative contrastive learning (RQ4)

To illustrate the efficacy of our design collaborative contrastive learning in alleviating the data sparsity issue we conduct a comprehensive series of experiments. In the Tmall dataset, we categorized users into five groups based on the sparsity of their target behavior (purchase). The X-axis delineates different sparsity levels of user groups, the left Y-axis denotes the number of users in each group, and the right Y-axis showcases the HR@20 values in the test results, presented in Fig. 6a. At all sparsity levels, MBCCL outperforms other multi-behavior and contrastive learning recommendation methods, demonstrating remarkable recommendation performance, particularly for users with sparse target behavior data.

Concurrently, with regard to the selection method for positive and negative sample pairs in collaborative contrastive learning, we conduct experiments where we omitted initial preference contrastive learning and final preference contrastive learning, utilizing two variants illustrated in the figure, as shown in Fig. 6b. "CML": is modeled after CML [16], where we defined positive and negative samples in contrastive learning as the target behavior and auxiliary behavior.

Fig. 6 Collaborative contrastive learning impact studies



(a) Performance w.r.t different sparsity degrees of interaction data for users and items, respectively, on Tmall dataset.

(b) Performance comparison between different methods for constructing positive and negative pairs

"S-MBRec": is modeled after S-MBRec [35], where we set positive and negative pairs as purchase behavior and other single behaviors. It is evident that our collaborative contrastive learning outperforms these variants, underscoring that our coordinated approach involving temporal and attribute information, by contrasting users' multi-behavior and attribute information at different times and employing diverse methods for selecting positive and negative samples, this approach effectively mitigates data sparsity issues, enhancing the diversity and fairness of the model's recommendations.

Hyperparameter study (RQ5)

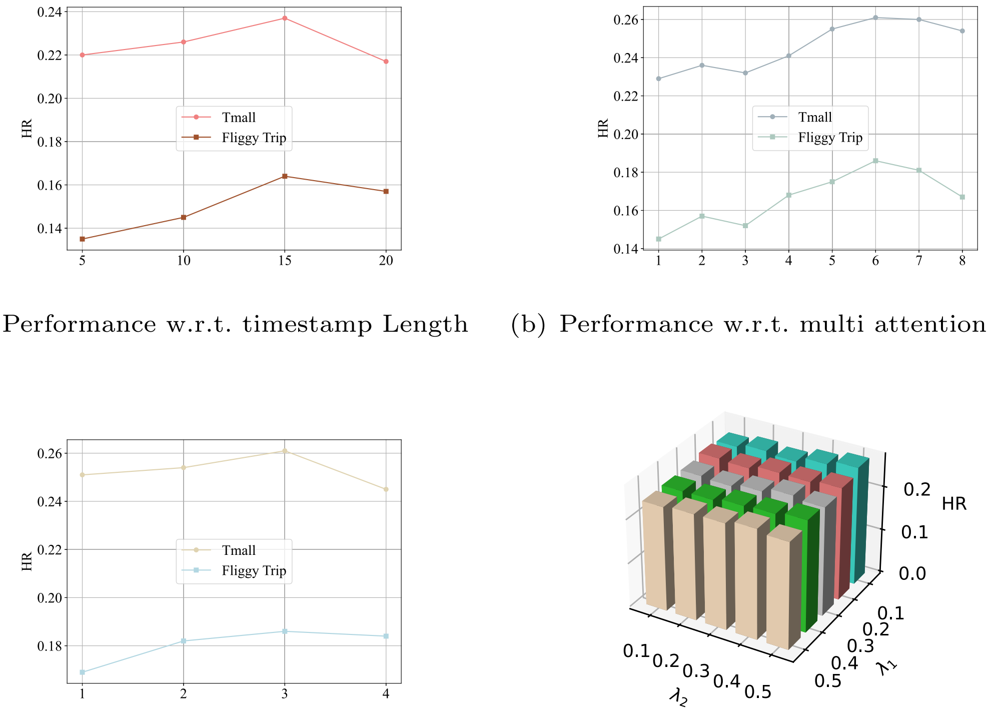
To evaluate the performance of our proposed MBCCL model, we utilize four experiments to explore the impact of various parameter settings. The model incorporates four key parameter configurations: the length of timestamps $t_{u,i,b}$, the number of attention heads in temporal aware attention, the number of layers in the light graph convolutional neural network l, and the weight coefficients λ_1 λ_2 in the joint loss function. The conclusions drawn from these experiments are as follows:

Impact of timestamp length

In the propose temporal aware encoding module, we explored the impact of varying lengths of timestamps $t_{u,i,b}$ on the corresponding HR scores. Figure 7a illustrates the results of this analysis, revealing the following trends: The temporal aware encoding module attains higher HR scores on both datasets when the timestamp length is set to 10. However, with an increase in the timestamp length, the module's performance begins to decline. This phenomenon may be attributed to the fact that projects with interactions that occurred a long time ago may not effectively reflect the user's current preferences.

Impact the number of attention heads

In temporal aware attention, we investigated the influence of employing different numbers of attention heads on HR and NDCG scores. Figure 7b presents the results of this analysis, revealing the following trend: The attribute weight enhancement module achieves higher HR and NDCG scores on both datasets when the number of attention heads is set to 6.



(a) Performance w.r.t. timestamp Length

(b) Performance w.r.t. multi attention heads

(c) Performance w.r.t. number of light graph convolutional neural network layers

(d) Performance w.r.t. weight coefficients

Fig. 7 Parameters' analysis

Impact of the number of light graph convolutional neural network layers

In the MBCCL model, we examine the impact of using different numbers of LGCN layers on HR and NDCG scores. Figure 7c displays the results of this analysis, and we observed the following trends: Using three layers of the light graph convolutional neural network effectively improves the model's performance, demonstrating that embedding with higher order connectivity has a positive effect. However, as the number of layers increases, it introduces excessive noise and leads to the problem of over-smoothing in the propagation layers, causing a decline in the model's performance.

Impact of weight coefficients

From Fig. 7d, it is evident that the model achieves its best performance when the weight for the final preference embedding contrastive loss λ_2 is set to 0.5, and the weight for the initial preference λ_1 is set to 0.1. This observation implies the following: User preferences at the final moments strongly correlate with the target behavior, such as purchasing. Conversely, at the initial stages of user interactions, preferences are more diverse and less predictable. Therefore, allocating a lower weight to the initial preference in the loss function recognizes this diversity. Striking the right balance between these weights enables the model to adeptly capture both the evolving preferences over time and the diverse initial preferences, resulting in enhanced performance in recommendation tasks.

Conclusion

In this work, MBCCL integrates users' multi-behavior interactions as auxiliary information, enabling the model to acquire a more profound understanding of diverse user preferences during interactions with items. Furthermore, we introduce the temporal aware encoding module and the attribute weight enhancement module, incorporate two attention mechanisms, and employ LGCN to skillfully encode comprehensive user and item attribute information and temporal data from diverse perspectives. In addition, to address data sparsity issues in the target behavior and enhance the diversity and fairness of model recommendations, we introduce collaborative contrastive learning (CCL), which collaboratively captures users' multi-behavior personalized preferences from both temporal and attribute perspectives. The experiments conduct on two datasets demonstrate that the model significantly outperforms various baseline recommendation models.

Sequential recommendation models employ contrastive learning based on heuristic methods to augment the data. However, these approaches may fall short of effectively preserving the inherent semantic structure of sequences, making them susceptible to noise interference. Furthermore, user interaction data are fraught with substantial biases, including selection bias, position bias, exposure bias, and popularity bias. Blindly fitting data without accounting for these inherent biases can compromise the interpretability and robustness of the model's recommendations. In our future work, we intend to explore the integration of causal learning into our collaborative contrastive learning models to address bias issues. Concurrently, we will investigate enhanced data augmentation methods that bolster the model's capacity to capture user preferences while preserving the semantic structure of sequences.

Acknowledgements This work was supported by National Natural Science Foundation of China grant no. 62141201, National Natural Science Foundation of Chongqing grant no. CSTB2022NSCQ-MSX1672, and Chongqing University of Technology Graduate Education Quality Development Action Plan Funding Results no. gzlcx20233193.

Declarations

Conflict of 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.

References

1. He X, Liao L, Zhang H, Nie L, Hu X, Chua T-S (2017) Neural collaborative filtering. In: Proceedings of the 26th international conference on world wide web, pp 173–182
2. Gu Y, Ding Z, Wang S, Yin D (2020) Hierarchical user profiling for e-commerce recommender systems. In: Caverlee J, Hu XB, Lalmas M, Wang W (eds) WSDM '20: the thirteenth ACM international conference on web search and data mining, Houston, TX, USA, February 3–7, 2020. ACM, pp 223–231
3. Gao G, Bao Z, Cao J, Qin AK, Sellis T, Wu Z (2019) Location-centered house price prediction: A multi-task learning approach. CoRR arXiv:1901.01774
4. Sun F, Liu J, Wu J, Pei C, Lin X, Ou W, Jiang P (2019) Bert4rec: sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM international conference on information and knowledge management, pp 1441–1450

5. Sarwar BM, Karypis G, Konstan JA, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Shen VY, Saito N, Lyu MR, Zurko ME (eds) Proceedings of the tenth international world wide web conference, WWW 10, Hong Kong, China, May 1–5, 2001. ACM, pp 285–295. https://doi.org/10.1145/371920.372071
6. Koren Y, Bell RM, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42(8):30–37
7. Zhu G, Cao J, Chen L, Wang Y, Bu Z, Yang S, Wu J, Wang Z (2023) A multi-task graph neural network with variational graph autoencoders for session-based travel packages recommendation. ACM Trans Web 17(3):18–11830. https://doi.org/10.1145/3577032
8. Wu Z, Pan S, Chen F, Long G, Zhang C, Philip SY (2020) A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst 32(1):4–24
9. Tang J, Wang K (2018) Personalized top-n sequential recommendation via convolutional sequence embedding. In: Chang Y, Zhai C, Liu Y, Maarek Y (eds) Proceedings of the eleventh ACM international conference on web search and data mining, WSDM 2018, Marina Del Rey, CA, USA, February 5–9, 2018. ACM, pp 565–573. https://doi.org/10.1145/3159652.3159656
10. Nguyen J, Zhu M (2013) Content-boosted matrix factorization techniques for recommender systems. Stat Anal Data Min 6(4):286–301
11. Wang Z, Lin G, Tan H, Chen Q, Liu X (2020) CKAN: collaborative knowledge-aware attentive network for recommender systems. In: Huang JX, Chang Y, Cheng X, Kamps J, Murdock V, Wen J, Liu Y (eds) Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020. ACM, pp 219–228
12. Liu F, Cheng Z, Zhu L, Gao Z, Nie L (2021) Interest-aware message-passing gcn for recommendation. In: Proceedings of the web conference 2021, pp 1296–1305
13. Fang H, Zhang D, Shu Y, Guo G (2020) Deep learning for sequential recommendation: algorithms, influential factors, and evaluations. ACM Trans Inf Syst (TOIS) 39(1):1–42
14. Wang W, Zhang W, Liu S, Liu Q, Zhang B, Lin L, Zha H (2020) Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction. In: Proceedings of the web conference 2020, pp 3056–3062
15. Xia L, Huang C, Xu Y, Pei J (2022) Multi-behavior sequential recommendation with temporal graph transformer. IEEE Trans Knowl Data Eng
16. Capinding AT (2021) Effect of teams-games tournament (TGT) strategy on mathematics achievement and class motivation of grade 8 students. Int J Game Based Learn 11(3):56–68. https://doi.org/10.4018/IJGBL.2021070104
17. Wei W, Huang C, Xia L, Xu Y, Zhao J, Yin D (2022) Contrastive meta learning with behavior multiplicity for recommendation. In: Candan KS, Liu H, Akoglu L, Dong XL, Tang J (eds) WSDM '22: the fifteenth ACM international conference on web search and data mining, virtual event/Tempe, AZ, USA, February 21–25, 2022. ACM, pp 1120–1128. https://doi.org/10.1145/3488560.3498527
18. Wu J, Wang X, Feng F, He X, Chen L, Lian J, Xie X (2020) Self-supervised graph learning for recommendation. arXiv:2010.10783
19. Wu Y, Xie R, Zhu Y, Ao X, Chen X, Zhang X, Zhuang F, Lin L, He Q (2022) Multi-view multi-behavior contrastive learning in recommendation. In: Bhattacharya A, Lee J, Li M, Agrawal D, Reddy PK, Mohania MK, Mondal A, Goyal V, Kiran RU (eds) Database systems for advanced applications—27th international conference, DASFAA 2022, Virtual Event, April 11–14, 2022, Proceedings, Part II. Lecture Notes in Computer Science, vol 13246. Springer, pp 166–182. https://doi.org/10.1007/978-3-031-00126-0_11
20. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. CoRR arXiv:1706.03762
21. Tian Y, Chang J, Niu Y, Song Y, Li C (2022) When multi-level meets multi-interest: A multi-grained neural model for sequential recommendation. In: Amigó E, Castells P, Gonzalo J, Carterette B, Culpepper JS, Kazai G (eds) SIGIR '22: the 45th international ACM SIGIR conference on research and development in information retrieval, Madrid, Spain, July 11–15, 2022. ACM, pp 1632–1641. https://doi.org/10.1145/3477495.3532081
22. Zhao K, Zhao X, Zhang Z, Li M (2022) Mae4rec: Storage-saving transformer for sequential recommendations. In: Hasan MA, Xiong L (eds) Proceedings of the 31st ACM international conference on information & knowledge management, Atlanta, GA, USA, October 17–21, 2022. ACM, pp 2681–2690. https://doi.org/10.1145/3511808.3557461
23. Zhu T, Sun L, Chen G (2023) Graph-based embedding smoothing for sequential recommendation. IEEE Trans Knowl Data Eng 35(1):496–508. https://doi.org/10.1109/TKDE.2021.3073411
24. Dong Y, Zha Y, Zhang Y, Zha X (2023) Long- and short-term collaborative attention networks for sequential recommendation. J Supercomput 79(16):18375–18393. https://doi.org/10.1007/S11227-023-05348-3
25. Su J, Chen C, Lin Z, Li X, Liu W, Zheng X (2023) Personalized behavior-aware transformer for multi-behavior sequential recommendation. In: El-Saddik A, Mei T, Cucchiara R, Bertini M, Vallejo DPT, Atrey PK, Hossain MS (eds) Proceedings of the 31st ACM international conference on multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023–3 November 2023. ACM, pp 6321–6331. https://doi.org/10.1145/3581783.3611723
26. Shen Y, Ou B, Li R (2022) MBN: towards multi-behavior sequence modeling for next basket recommendation. ACM Trans Knowl Discov Data 16(5):81–18123. https://doi.org/10.1145/3497748
27. Yang Y, Huang C, Xia L, Liang Y, Yu Y, Li C (2022) Multi-behavior hypergraph-enhanced transformer for sequential recommendation. In: Zhang A, Rangwala H (eds) KDD '22: The 28th ACM SIGKDD conference on knowledge discovery and data mining, Washington, DC, USA, August 14–18, 2022. ACM, pp 2263–2274. https://doi.org/10.1145/3534678.3539342
28. Yuan E, Guo W, He Z, Guo H, Liu C, Tang R (2022) Multi-behavior sequential transformer recommender. In: Amigó E, Castells P, Gonzalo J, Carterette B, Culpepper JS, Kazai G (eds) SIGIR '22: the 45th international ACM SIGIR conference on research and development in information retrieval, Madrid, Spain, July 11–15, 2022. ACM, pp 1642–1652. https://doi.org/10.1145/3477495.3532023
29. Ning Y, Peng J, Liu Q, Huang Y, Sun W, Du Q (2023) Contrastive learning based on category matching for domain adaptation in hyperspectral image classification. IEEE Trans Geosci Remote Sens 61:1–14. https://doi.org/10.1109/TGRS.2023.3295357
30. Lee S, Lee M (2023) Enhancing text comprehension for question answering with contrastive learning. In: Can B, Mozes M, Cahyawijaya S, Saphra N, Kassner N, Ravfogel S, Ravichander A, Zhao C, Augenstein I, Rogers A, Cho K, Grefenstette E, Voita L (eds) Proceedings of the 8th workshop on representation learning for NLP, RepL4NLP@ACL 2023, Toronto, Canada, July 13, 2023. Association for Computational Linguistics, pp 75–86. https://doi.org/10.18653/V1/2023.REPL4NLP-1.7
31. Cai X, Huang C, Xia L, Ren X (2023) Lightgcl: simple yet effective graph contrastive learning for recommendation. In: The eleventh international conference on learning representations, ICLR 2023, Kigali, Rwanda, May 1–5, 2023. OpenReview.net. https://openreview.net/pdf?id=FKXVK9dyMM
32. Wang Z, Liu H, Chen D (2022) Multi-level contrastive learning framework for sequential recommendation. In: Proceedings of the 31st ACM international conference on information & knowledge management, Atlanta, GA, USA, October 17–21, 2022. ACM, pp 2098–2107
33. Wang L, Zhao T (2022) Explanation guided contrastive learning for sequential recommendation. In: Proceedings of the 31st ACM inter-

national conference on information & knowledge management, Atlanta, GA, USA, October 17–21, 2022. ACM, pp 2017–2027

34. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Teh YW, Titterington DM (eds) Proceedings of the thirteenth international conference on artificial intelligence and statistics,2010, Chia Laguna Resort, Sardinia, Italy, May 13–15, 2010. JMLR Proceedings, vol 9. JMLR.org, pp 249–256
35. Peng Z, Liu H, Jia Y, Hou J (2021) Attention-driven graph clustering network. In: Shen HT, Zhuang Y, Smith JR, Yang Y, César P, Metze F, Prabhakaran B (eds) MM '21: ACM Multimedia Conference, Virtual Event, China, October 20–24, 2021. ACM, pp 935–943
36. He X, Deng K, Wang X, Li Y, Zhang Y, Wang M (2020) Lightgcn: simplifying and powering graph convolution network for recommendation. CoRR arXiv:2002.02126
37. Zheng Y, Gao C, Chang J, Niu Y, Song Y, Jin D, Li Y (2022) Disentangling long and short-term interests for recommendation. In: Laforest F, Troncy R, Simperl E, Agarwal D, Gionis A, Herman I, Médini L (eds) WWW '22: The ACM web conference 2022, virtual event, Lyon, France, April 25–29, 2022. ACM, pp 2256–2267. https://doi.org/10.1145/3485447.3512098
38. Gu S, Wang X, Shi C, Xiao D (2022) Self-supervised graph neural networks for multi-behavior recommendation. In: Raedt LD (ed) Proceedings of the thirty-first international joint conference on artificial intelligence, IJCAI 2022, Vienna, Austria, 23–29 July 2022. ijcai.org, pp 2052–2058
39. van den Oord A, Li Y, Vinyals O (2018) Representation learning with contrastive predictive coding. CoRR arXiv:1807.03748
40. Yang Z, Ding M, Xu B, Yang H, Tang J (2022) STAM: A spatiotemporal aggregation method for graph neural network-based recommendation. In: Laforest F, Troncy R, Simperl E, Agarwal D, Gionis A, Herman I, Médini L (eds) WWW '22: the ACM web conference 2022, Virtual Event, Lyon, France, April 25–29, 2022. ACM, pp 3217–3228. https://doi.org/10.1145/3485447.3512041
41. Chen C, Zhang M, Zhang Y, Ma W, Liu Y, Ma S (2020) Efficient heterogeneous collaborative filtering without negative sampling for recommendation. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 19–26
42. Chen C, Ma W, Zhang M, Wang Z, He X, Wang C, Liu Y, Ma S (2021) Graph heterogeneous multi-relational recommendation. In: Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence, EAAI 2021, Virtual Event, February 2–9, 2021. AAAI Press, pp 3958–3966
43. Jin B, Gao C, He X, Jin D, Li Y (2020) Multi-behavior recommendation with graph convolutional networks. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, pp 659–668
44. Xia L, Xu Y, Huang C, Dai P, Bo L (2021) Graph meta network for multi-behavior recommendation. In: SIGIR '21: the 44th international ACM SIGIR conference on research and development in information retrieval, virtual event, Canada, July 11-15, 2021. ACM, pp 757–766
45. Sun F, Liu J, Wu J, Pei C, Lin X, Ou W, Jiang P (2019) Bert4rec: sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM international conference on information and knowledge management, CIKM 2019, Beijing, China, November 3–7, 2019. ACM, pp 1441–1450
46. Wang S, Li X, Kou X, Zhang J, Zheng S, Wang J, Gong J (2021) Sequential recommendation through graph neural networks and transformer encoder with degree encoding. Algorithms 14(9):263. https://doi.org/10.3390/a14090263
47. Wang X, He X, Cao Y, Liu M, Chua T-S (2019) Kgat: knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 950–958