Abstract In real recommendation scenarios, users often have different types of behaviors, such as clicking and buying. Existing research methods show that it is possible to capture the heterogeneous interests of users through different types of behaviors. However, most multi-behavior approaches have limitations in learning the relationship between different behaviors. In this paper, we propose a novel multilayer perceptron (MLP)-based heterogeneous sequential recommendation method, namely behavior-aware multilayer perceptron (BMLP). Specifically, it has two main modules, including a heterogeneous interest perception (HIP) module, which models behaviors at multiple granularities through behavior types and transition relationships, and a purchase intent perception (PIP) module, which adaptively fuses subsequences of auxiliary behaviors to capture users' purchase intent. Compared with mainstream sequence models, MLP is competitive in terms of accuracy and has unique advantages in simplicity and efficiency. Extensive experiments show that BMLP achieves significant improvement over state-of-the-art algorithms on four public datasets. In addition, its pure MLP architecture leads to a linear time complexity.

Keywords Sequential Recommendation; Heterogeneous Behaviors; Multilayer Perceptron

1 Introduction

Recommender systems can effectively alleviate the problem of information overload, particularly with the substantial growth of online users, items, and information in recent years. However, traditional recommendation methods cannot capture users' dynamic interests and intent. Sequential recommendation methods, which aim to model the behavior sequence of users to obtain more accurate, dynamic, and personalized recommendation lists, overcome this problem well.

Recently, deep learning-based sequential recommendation methods mine users' latent representations and complex sequential relationships from the interaction data. RNN is proposed to capture the sequential information between items [1, 2]. CNN-based methods [3, 4] use filters to learn users' short-term interests. After that, attention networks became popular in many fields, e.g., being used to learn the relevant weights between items [5–7]. GNN-based methods [8–11] structure sessions into graphs to capture richer relevance of items. Those mainstream deep learning-based methods have achieved exciting results, gradually dominating the recommender systems. However, they have a limitation in common, i.e., they can only deal with a single type of behavior.

In a real scenario, the behaviors of users are often rich and diverse, which is unreasonable to regard them as the same behavior type or only select the behaviors of one single type such as purchase to learn the users' preferences. There are two categories of works on modeling multiple behaviors. Heterogeneous recommendations [12–14] aggregate representations of various behaviors as overall user preferences. They ignore the dynamic dependencies between items and behaviors. Heterogeneous sequential recommendation is an emerging and significant research problem that has relatively few works in its field. Specifically, RNN-based methods [15–17] distinguish the feedback of different behaviors by introducing behavior types or integrating the behavior information into the RNN module. The Transformer-based method [18] obtains multiple interests of the users by modeling the subsequences of the same type separately. The GNN-based method [19] captures relationships between the behaviors of the same type by constructing a global graph. These approaches may ignore complex behavior dependencies and transition relationships. Meanwhile, in e-commerce websites, the purchase behavior mainly depends on the overall user preferences and some recent browsing or favoriting behaviors. However, for the aforementioned heterogeneous recommendation methods (e.g., MBGMN [12], MBGCN [13]), they fuse multiple behaviors to obtain the static interest of users, which only considers the overall user preferences. The previous works for the heterogeneous sequential recommendation can be broadly divided into two branches. One is to model the entire heterogeneous sequence while introducing the behavior information [15–17], which may not take adequate advantage of recent auxiliary behaviors to capture the intent of the users. The other is to disrupt the integrity of the sequences (e.g., MGNN [19], DMT [18], SDM [20]), modeling the single behavior subsequences separately, which may have a bias in capturing the overall interest of the users. Therefore, we consider some specific challenges of heterogeneous sequential recommendation: (i) The complexity of heterogeneous sequential recommendation models. Almost all the works utilize RNN, Transformer or GNN to capture

sequential patterns. (ii) The complex dependencies of behavior information. Distinguishing the feedback of different behaviors and the dependencies between behaviors is critical for accurately learning a user's preferences. (iii) The uncertainty of a user's intents. In a heterogeneous sequence of intertwined target and auxiliary behaviors, it is difficult to accurately predict what the user is likely to purchase.

To address the above three challenges, we propose a novel heterogeneous sequential recommendation method, i.e., behavior-aware MLP (BMLP), consisting of a heterogeneous interest perception (HIP) module and a purchase intent perception (PIP) module. (i) We use a pure MLP architecture, which has a lower time complexity. (ii) The heterogeneous interest perception (HIP) module performs multi-granularity processing of the entire heterogeneous sequence by introducing behavior types and behavior transition relations. It captures the dependencies between behaviors more comprehensively. (iii) The purchase intent perception (PIP) module adaptively fuses the auxiliary behaviors such as clicks. It captures the potential purchase intent of users more precisely. We then conduct extensive experiments on four datasets and find that our BMLP can beat the current mainstream state-of-the-art baselines. We summarize the main contributions of this work as follows.

- We propose a novel pure MLP-based recommendation approach that is simple and efficient. To the best of our knowledge, this is the first work using MLP to tackle heterogeneous sequential recommendation.
- Our HIP module captures the interaction patterns between behaviors from a multi-granularity perspective, namely behavior types and behavior transition relations. Meanwhile, our PIP module aggregates multiple auxiliary behavior subsequences. These two well-designed modules fit well with a user's purchasing habits.
- Compared with the existing state-of-the-art baselines, our BMLP achieves significant improvement on four public datasets.

2 Related Work

In this section, we review and summarize some related works on factorization-based and deep learning-based recommendation.

2.1 Factorization-based Recommendation

In the early studies on sequential recommendation, FPMC [21] integrates the sequential information for personalization and uses Markov chains to capture the first-order relationships between items. Because the first-order relationship is too simplified, Fossil [22] extends it to a multi-order dependency. Later, TransRec [23] embeds items as points in a translation space and models users as translation vectors existing in that space.

BF [24] uses matrix factorization techniques to model each behavior separately, decomposing users' interest profiles into multiple behavior profiles. Based on TransRec [23], TransRec++ [25] introduces behavior transition vectors to further characterize the dependencies between behaviors.

2.2 Deep Learning-based Recommendation

Sequential Recommendation. In recent years, deep learning has rapidly become popular in sequential recommendation due to its strong modeling ability and generalization. With the introduction of deep learning models, the performance has been further improved. The recurrent neural network (RNN) based method GRU4Rec [1] consumes the sequential information and captures the user's dynamic intent very well. The convolutional neural network (CNN) based method Caser [3] utilizes some horizontal and vertical filters to search for local information in sequences and global representations of users to record the long-term interests. The attention-based method SASRec [5] emphasizes the important items and downplays the irrelevant items by the attention mechanism. The graph neural network (GNN) based method SRGNN [8] captures complex transition relationships between items by constructing a graph from a sequence. Recently, with the advances in MLP architecture [26], MLP-based recommendation models are springing up. MOI-Mixer [27] applies this model to sequential recommendation for the first time. FMLP [28] introduces a filter layer based on the MLP architecture. MLP4Rec [29] expands the input tensor into 3-D by introducing the item properties.

Heterogeneous Non-sequential Recommendation. Many deep learning techniques (e.g., MLP, autoencoder, and GNNs) have been widely adopted in multi-behavior recommendation and have achieved remarkable performance. EHCF [32] captures the complex relations between different behaviors by relating the transition of each behavior.

VAE++ [14] is a VAE-based method which designs two representation enhancement modules to capture multiple behavior signals. For the GNN-based methods in multi-behavior recommendation, MBGCN [13] designs a unified graph for representing multi-behavior data and performs graph convolution operation to learn node representations under different behaviors. MBGMN [12] proposes a meta-graph neural network to capture the complicated dependencies across different types of user-item interactions with the meta-learning paradigm. CRGCN [33] adopts a cascading GCN structure to learn users' preferences under each behavior. To alleviate the data sparsity, some recent works are trying to leverage contrastive learning. CML [36] designs a multi-behavior contrastive learning paradigm to capture relations across different behaviors. MMCLR [37] proposes a multi-behavior contrastive learning task and a multi-view contrastive learning task. MixMBR [38] introduces a mixup data augmentation method and combines it with contrastive learning. These methods do not introduce sequential information and differ significantly from our problem definition, for which reason we do not include them in the empirical studies.

Heterogeneous Sequential Recommendation. Existing heterogeneous sequential recommendation algorithms are still relatively few. RLRL [17] uses the behavior transition matrix to represent the relationship between behaviors and uses RNN and log-bilinear to capture a user's interests. RIB [15] also introduces the behavior types and incorporates them into the GRU module. BINN [16] introduces the behavior information inside LSTM [30] and learns both the long-term and short-term interests using users' past behaviors and items. DMT [18] uses Transformer to model the subsequences of each same type of behavior, which captures the interactions between the same behaviors separately. MGNN-SPred [19] constructs heterogeneous graphs about multiple types of behaviors. MSR [31] uses GNN to model item sequences and GRU to model behavior sequences. We can see that these methods are based on RNN, GNN, and Transformer. Different from the above methods, we propose a heterogeneous sequential recommendation model based on MLP.

3 METHODOLOGY

In this section, we first formally describe the studied problem and then introduce our BMLP in detail.

3.1 Problem Definition

In heterogeneous sequential recommendation (HSR), our task is to predict the next item likely to be purchased for each user based on a user's historical interaction sequence with different types of behaviors, such as clicks and purchases. In contrast to multi-task recommendation [34], which aims for balanced performance improvement across all behaviors, our goal is to predict the purchase behavior. Without loss of generality, we have some different types of behaviors $b \in \mathcal{B}$ given by a set of users $\mathcal{U}$ to a set of items $\mathcal{I}$. In HSR, we define a historical heterogeneous sequence of a user u as $\mathcal{S}_u = \{(i_u^1, b_u^1), (i_u^2, b_u^2), \ldots, (i_u^{|\mathcal{S}_u|}, b_u^{|\mathcal{S}_u|})\}$, $i_u^{\cdot} \in \mathcal{I}$; $b_u^{\cdot} \in \mathcal{B}$. Our goal is to predict the next purchased item $i_u^{|\mathcal{S}_u|+1}$ of a user u from $\mathcal{U}$ as accurately as possible.

3.2 Overview of Behavior-Aware MLP

The overall structure of BMLP is shown in Fig 1 and consists of three modules as follows: 1) The HIP module encodes the entire heterogeneous sequence to ensure the integrity of the sequential information and capture fine-grained dependencies. It mainly learns the dependencies between the items and the representation information of items and behavior types separately by transposing the input tensor and using MLP twice. 2) The PIP module also firstly transposes each auxiliary behavior subsequence, then uses an MLP to learn the dependencies between the items of each subsequence separately, and finally uses an MLP to fuse all the auxiliary behavior embeddings. 3) The recommendation module adaptively fuses the learned interest and intent from two complementary modules.

As shown in Fig 1, we first encode the item embeddings and behavior embeddings of the entire heterogeneous sequence and each auxiliary behavior subsequence. For the entire heterogeneous sequence, we further introduce behavior transition embedding [25]. Then, we use the HIP module and PIP module to compute the user's overall heterogeneous interest and purchase intent, respectively. Finally, we fuse them using a gating component to obtain the final interest. We then use this final interest to calculate the dot product with the item embeddings, resulting in the corresponding item rating predictions for each user.

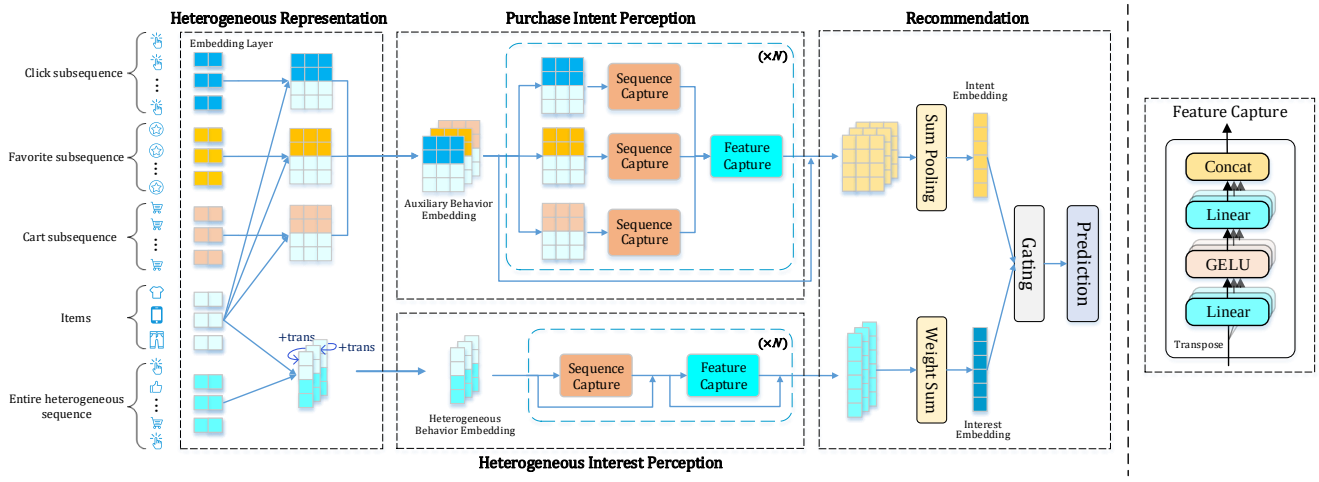


Fig. 1: The structure of the Behavior-aware MLP (BMLP) is divided into Heterogeneous Interest Perception (HIP), Purchase Intent Perception (PIP), and recommendation module. Details of the behavior replacement block and the Feature Capture Block (FCB) are shown in the right panel. The Sequence Capture Block (SCB) is a special case of FCB.

3.3 Heterogeneous Interest Perception

By modeling the entire heterogeneous sequence, it is possible to perceive the overall preferences of the user. However, this may suffer from insufficient utilization of different behavior information. To address this, we introduce the behavior type information and the behavior transition relations in the user sequences.

First of all, we select the last L (item, behavior) pairs for each user, which is formalized as $\mathcal{S}_u = \{(i_u^1, b_u^1), (i_u^2, b_u^2), \ldots, (i_u^L, b_u^L)\}$. Notice that L is usually chosen to be a relatively large value, e.g., $L = 50$, which is determined according to the size and distribution of a dataset. Moreover, we will pad items or behaviors to the beginning of a sequence if the length is shorter than L.

For each item i_u^t and the corresponding behavior b_u^t of a user u in an interaction sequence $\mathcal{S}_u$, since we deal with a sequence with different types of behaviors, in order to distinguish the behavior types of the items that a user interacts with, we use $B_b \in \mathbb{R}^{1\times d}$ to represent the embedding of behavior type b. We assume that the behavior types of two adjacent items in a sequence are dependent. Following [25], we use $\text{trans}(b_u^t, b_u^{t+1}) \in \mathbb{R}^{1\times d}$ to denote the transition relationship between two consecutive behaviors. Then, we combine the behavior embedding and the behavior transition embedding:

$$M_u^t = B_{b_u^t} + \text{trans}(b_u^t, b_u^{t+1}) \tag{1}$$

Next, we use $V_{i_u^t}$ to represent the embedding of item i_u^t. Finally, we concatenate the item embedding and the behavior embedding to obtain a heterogeneous embedding $\mathbf{x}_u^t \in \mathbb{R}^{1\times 2d}$:

$$\mathbf{x}_u^t = \text{concat}(M_u^t, V_{i_u^t}) \tag{2}$$

After the above process, we obtain an input matrix $\mathbf{X}_u^{(0)} = [\mathbf{x}_u^1; \ldots; \mathbf{x}_u^L] \in \mathbb{R}^{L\times 2d}$, where L is the length of the interaction sequence. Notice that we now omit the subscript u in $\mathbf{X}_u^{(0)}$ for brevity. Then, we feed the input matrix $\mathbf{X}^{(0)}$ into a series of stacked blocks. First of all, we feed it into a sequence capture block (SCB) used to capture the sequential information between items, where the output of the n-th SCB is as follows:

$$\mathbf{X}_{\text{SCB}}^{(n-1)} = \mathbf{X}^{(n-1)} + \text{SCB}\left(\text{LayerNorm}(\mathbf{X}^{(n-1)})^T\right)^T \tag{3}$$

$$\text{SCB}(\mathbf{X}^T) = \text{GELU}(\mathbf{X}^T W_{\text{SCB}}^1) W_{\text{SCB}}^2 \tag{4}$$

where $W_{\text{SCB}}^1 \in \mathbb{R}^{L\times d_t}$ and $W_{\text{SCB}}^2 \in \mathbb{R}^{d_t\times L}$ are trainable matrices, and d_t represents the hidden layer dimension in SCB.

After updating the columns of the input matrix $\mathbf{X}^{(0)}$, we capture the sequential information between the items from another perspective. Notice that each column of $\mathbf{X}^{(0)}$ corresponds to L items in a channel. Therefore, it can perceive the positional relationship between items.

We use $\mathbf{X}_{\text{SCB}}^{(n-1)}$ as input to the feature capture block (FCB), where $\mathbf{X}_{\text{SCB}}^{(n-1)}$ is divided into the H parts of the hidden layer dimension, i.e., $\mathbf{X}_{(h-1)\times\frac{2d}{H}+1:h\times\frac{2d}{H}}$, $h = 1, 2, \ldots, H$. It guides

the model to focus on the information in different spaces. We feed $\mathbf{X}_{(h-1)\times\frac{2d}{H}+1:h\times\frac{2d}{H}} \in \mathbb{R}^{L\times\frac{2d}{H}}$ into a smaller feature capture block, defined as a head. Finally, we merge and fuse the information of those H heads. Among them, the n-th block is calculated as follows:

$$\mathbf{X}^{(n)} = \mathbf{X}_{\text{SCB}}^{(n-1)} + \text{FCB}(\text{LayerNorm}(\mathbf{X}_{\text{SCB}}^{(n-1)})) \tag{5}$$

$$\text{FCB}(\mathbf{X}) = \text{concat}(\ldots, \text{head}_h, \ldots)W_{\text{FCB}}^{O} \tag{6}$$

$$\text{head}_h = \sigma(\mathbf{X}_{(h-1)\times\frac{2d}{H}+1:h\times\frac{2d}{H}} W_{\text{FCB}}^{1})W_{\text{FCB}}^{2}, h = 1, 2, \ldots, H \tag{7}$$

where $W_{\text{FCB}}^{O} \in \mathbb{R}^{2d\times 2d}$, $W_{\text{FCB}}^{1} \in \mathbb{R}^{\frac{2d}{H}\times d_c}$ and $W_{\text{FCB}}^{2} \in \mathbb{R}^{d_c\times\frac{2d}{H}}$ are learnable matrices, $\sigma(\cdot)$ is a sigmoid activation function, and d_c is the hidden layer dimension in FCB. We can see that SCB is a special case of FCB when $H = 1$. Notice that SCB and FCB deal with the sequential information and the feature information, respectively. We take the output matrix $\mathbf{X}^{(N)} \in \mathbb{R}^{L\times 2d}$. Considering the different importance of each item in a sequence, we use a weighting mechanism to aggregate them:

$$\alpha_t = \frac{\exp(\mathbf{x}_t^{(N)} W_\alpha)}{\sum_{j=1}^{L} \exp(\mathbf{x}_j^{(N)} W_\alpha)} \tag{8}$$

where $W_\alpha \in \mathbb{R}^{2d\times 1}$ is a learnable vector, and $\alpha_t \in \mathbb{R}$ represents the weight for the t-th item in the sequence. Then, the corresponding item embeddings are fused by the weights:

$$\mathbf{e}_g = \text{dropout}(\sum_{t=1}^{L} \alpha_t \mathbf{x}_t^{(N)}) \tag{9}$$

In this module, we obtain the corresponding heterogeneous behavior representations by fusing item representations, behavior representations, and behavior transition information. Then, we utilize the SCB and FCB modules to capture the fine-grained long-term interests of users.

3.4 Purchase Intent Perception

In the previous module, we fuse the behavior embedding and the item embedding to obtain the global heterogeneous interest of each user. However, there are some limitations in this approach. Firstly, the HIP module cannot distinguish the behavior type of the next predicted item. Although the long-term heterogeneous interest can help us understand a user's general preferences, it does not take into account the specificity of his or her purchase behavior. This is not consistent with our goal of predicting the next item that a user may purchase in the future. Secondly, in an e-commerce scenario, a user tends to click some similar items or mark them as favorite before purchasing. This means that a recent auxiliary behavior has a greater influence on a user's purchase behavior. To address this, we further design a purchase intent perception module. This module takes into account the recent auxiliary behaviors to better understand a user's current purchase intent.

The recent auxiliary behaviors (e.g., clicks, favorites) of a user reflect the short-term purchase intent of the user, so we choose the most recent L' items w.r.t. each auxiliary behavior as a subsequence, where L' is relatively small (e.g., $L' = 5$). Earlier auxiliary behaviors usually have little effect on the current purchase intent because the purchase has already occurred or the user's interest has shifted.

For each auxiliary behavior, we concatenate the embedding of the item and the behavior type as the input of the purchase intent perception module:

$$\mathbf{h}_{b_u}^{t} = \text{concat}(B_{b_u^t}, V_{i_u^t}) \tag{10}$$

where $B_{b_u^t} \in \mathbb{R}^{1\times d}$ represents the embedding of the auxiliary behavior b_u^t and $V_{i_u^t} \in \mathbb{R}^{1\times d}$ represents the embedding of the item i_u^t. So, we obtain an input tensor $\mathcal{H}_u^{(0)} = [\mathbf{H}_{b_u}^{1}; \ldots; \mathbf{H}_{b_u}^{m}] \in \mathbb{R}^{L'\times m\times 2d}$. Notice that m indicates the number of auxiliary behaviors. Similarly, we omit the subscript u in $\mathcal{H}_u^{(0)}$ for brevity.

Firstly, we transpose the embedding matrix for each auxiliary behavior. Then we feed them into SCB and FCB, where the output is as follows:

$$\mathcal{H}_s^{(n-1)} = \text{stack}\left(\text{SCB}\left(\mathbf{H}_{b_u^1}^{(n-1)^T}\right)^T, ..., \text{SCB}\left(\mathbf{H}_{b_u^m}^{(n-1)^T}\right)^T\right) \tag{11}$$

$$\mathcal{H}^{(n)} = \mathcal{H}^{(n-1)} + \text{FCB}(\text{LayerNorm}(\mathcal{H}_s^{(n-1)})) \tag{12}$$

where the SCB$(\cdot)$ and FCB$(\cdot)$ are the same as that in the heterogeneous interest learning module. The difference between the HIP module and the PIP module is the heterogeneity and length of the processed sequence. We aggregate the last item embedding of each auxiliary behavior as the short-term purchase intent:

$$\mathbf{e}_l = \text{mean}(\mathcal{H}_{L'}^{(N)}) \tag{13}$$

where $\mathbf{e}_l \in \mathbb{R}^{1\times 2d}$, and $\mathcal{H}_{L'}^{(N)} \in \mathbb{R}^{m\times 2d}$ denotes the last item representation of the auxiliary behaviors.

3.5 Gating

We combine the local purchase intent $\mathbf{e}_l$ and the global heterogeneous interest $\mathbf{e}_g$ as follows:

$$\mathbf{g} = \sigma(\mathbf{e}_g W_g + \mathbf{e}_l W_l + \mathbf{b}_g) \tag{14}$$

$$\mathbf{z} = \mathbf{g} \otimes \mathbf{e}_g + (1 - \mathbf{g}) \otimes \mathbf{e}_l \tag{15}$$

where $\otimes$ is the element-wise product, $W_g \in \mathbb{R}^{2d \times 2d}$, $W_l \in \mathbb{R}^{2d \times 2d}$ and $\mathbf{b}_g \in \mathbb{R}^{1 \times 2d}$ are learnable weights and biases, and $\sigma(\cdot)$ is a sigmoid activation function to constrain the value of each entry in $\mathbf{g} \in \mathbb{R}^{1 \times 2d}$ to $(0, 1)$. Finally, we use a fully connected layer to make the prediction:

$$\hat{r}_{t+1} = \text{softmax}(\mathbf{z} W_r + \mathbf{b}_r) \tag{16}$$

where $W_r \in \mathbb{R}^{2d \times |\mathcal{I}|}$ is a learnable matrix, $\mathbf{b}_r \in \mathbb{R}^{1 \times |\mathcal{I}|}$ is the bias, and $\hat{r}_{t+1} \in \mathbb{R}^{1 \times |\mathcal{I}|}$ contains the predicted score on each item.

3.6 Loss Function

The loss function of the model is as follows:

$$\mathcal{L} = -\sum_{s \in \mathcal{S}} \sum_{j \in \mathcal{I}} y_{sj} \log(\hat{r}_{t+1,j}) + (1 - y_{sj}) \log(1 - \hat{r}_{t+1,j}) \tag{17}$$

where $y_{sj} = 1$ only if an item j is a truly interacted item in the sequence s at timestamp $t + 1$, and $y_{sj} = 0$ otherwise.

3.7 Discussions

In this subsection, we discuss three existing MLP-based sequential recommendation methods.

- MOI-Mixer [27]: It is the first work to apply MLP-Mixer [26] in a recommender system, which introduces higher-order interactions in the MLP layer. Notice that we introduce the multi-head mechanism in the MLP layer to learn the feature information in different subspaces, and capture behavior information with different granularities at the same time.
- FMLP-Rec [28]: It is an improved method based on MLP-Mixer [26]. The difference is that it passes a filter layer after encoding the items, which reduces the noise. Our BMLP extracts user interest and intent from heterogeneous sequences, which can also reduce noise to a certain extent because of their complementarity.
- MLP4Rec [29]: It is also an improved method based on MLP-Mixer [26], which introduces attributes such as categories and brands of items and expands them into a new dimension. Specifically, the input of MLP-Mixer is a 2-D matrix, while the input of MLP4Rec is a 3-D tensor.

We can see that the MLP architecture has rarely been exploited in recommendation systems. Notice that the structure of MLP is insensitive to the sequential information. Our BMLP captures the sequential pattern and achieves better performance in sequential recommendation mainly with the following two reasons: 1) the sliding window training approach, which allows the model to capture the sequential information in the training process; and 2) the transposition of the input matrix, which allows the MLP to capture the dependencies between items.

4 Experiments

In this section, we study the effectiveness of our proposed model by conducting extensive experiments on four datasets. We first introduce the experimental setup in detail, including data processing, baselines, and evaluation metrics, and then focus on answering the following seven research questions.

- **RQ1** : How does our proposed model perform compared with the state-of-the-art models?
- **RQ2** : Why is our BMLP simpler and more efficient compared with the existing methods?
- **RQ3** : How will behavior diversity affect the performance of the model?
- **RQ4** : How do the components, i.e., FCB, SCB, and PIP, affect the performance of the model?
- **RQ5** : How does the PIP module perceives a user's purchase intent?
- **RQ6** : How do the heterogeneous non-sequential recommendation methods perform?
- **RQ7** : What is the impact of the values of the hyperparameters on the model?

4.1 Experimental Setting

4.1.1 Datasets

We conduct experiments on four datasets, i.e., RecSys Challenge 2015 (Rec15)[1], Tmall at IJCAI-15 Contest (Tmall)[2], MovieLens 1M (ML1M)[3], and Taobao user behaviors (UB)[4]. We process these datasets as follows: 1) For duplicated (user, item, behavior) records, we only keep the record with the earliest time. 2) For each user's interaction sequence, we use the last two purchases for validation and testing, respectively, and retain the auxiliary behaviors between these two purchases for final performance evaluation. 3) We remove the cold-start items in the validation and test sets.
Then we conduct some specific processing for each dataset.

Rec15. Rec15 was released by the RecSys 2015 competition, which contains 9,249,729 users, 52,739 items, and 34,154,697 interaction records. We preprocess it as follows: 1) Delete the items that have been purchased fewer than 5 times. 2) Delete the users who have purchased items fewer than 5 times.

Tmall. The Tmall dataset comes from the Tmall app and contains records of user interactions before and on the day of "Double 11". The original data contains 424,170 users, 1,090,390 items, four types of behaviors, and 54,925,330 interaction records. We preprocess it as follows: 1) Delete all the records on the day of "Double 11" to avoid impulsive consumption and the disruption caused by the promotional activities on that day. 2) Delete the items that have been purchased fewer than 20 times. 3) Delete the users who have purchased items fewer than 10 times.

ML1M. A dataset of movie ratings with 1 million ratings assigned by 6,000 users to 4,000 movies. We preprocess it as follows: 1) A rating of 5 is simulated as a purchase, and a rating smaller than 5 is taken as an auxiliary behavior [25]. 2) Delete the items that have been purchased fewer than 5 times. 3) Delete the users who have purchased items fewer than 5 times.

UB. A Taobao behavior dataset provided by Alibaba, which contains 987,994 users, 4,162,024 items, and 100,150,807 interaction records. We preprocess it as follows: 1) Delete the items that have been purchased fewer than 10 times. 2) Delete the users who have purchased items fewer than 5 times.

The statistics of the processed datasets are summarized in Table 1. In addition, we observe that some items have been examined in the input sequence while others have not. Considering the impact of the auxiliary behaviors on the target behavior and the fact that users in real-world recommendation scenarios are likely to purchase items they have previously clicked, the model recommends both examined and unexamined items. Subsequently, we conduct an in-depth analysis of this aspect in Section 4.7. The processed datasets, source code of our BMLP and the scripts used in the experiments are publicly available at https://csse.szu.edu.cn/staff/panwk/publications/BMLP/.

Table 1: Statistics of the four processed datasets used in the experiments.

Datasets	#Users	#Items	#Interactions	#Behaviors
Rec15	36,917	9,620	679,704	{Click, Buy}
Tmall	17,209	16,162	1,251,829	{Click, Favorite, Buy}
ML1M	5,645	2,357	885,598	{Click, Buy} (simulated)
UB	20,858	30,718	782,297	{Click, Cart, Favorite, Buy}

4.1.2 Evaluation Metrics

We adopt two widely used metrics to compare our proposed model with the baselines, i.e., hit ratio (HR@k) and normalized discounted cumulative gain (NDCG@k), where k is chosen from $\{10, 20\}$. HR considers whether the test items appear in a recommendation list, while NDCG is more concerned with the positions of the test items. For a heterogeneous sequence of a particular user, we take the last purchased item as the test data. According to the score of each item calculated in Eq.(16), we select k items with the highest scores as a recommendation list.

4.1.3 Baselines

We include some competitive baseline methods for both single-behavior sequential recommendation (i.e., Caser, GRU4Rec, SASRec, SRGNN and BERT4Rec) and heterogeneous sequential recommendation (i.e., RLBL, RIB, BINN, and MSR).

- Caser [3]: It constructs users' behavior embeddings into matrices and uses CNN to capture multi-order relationships between items.

[1] https://recsys.acm.org/recsys15/challenge/
[2] https://tianchi.aliyun.com/dataset/dataDetail?dataId=42
[3] https://grouplens.org/datasets/movielens/1m/
[4] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649

- GRU4Rec [1]: It takes a user's behaviors as a sequence and uses GRU to learn the dependencies between items.
- SASRec [5]: It is an attention-based sequential recommendation model that captures the importance of items in the sequence.
- SRGNN [8]: It captures complex transition relationships between the items by constructing a graph from a sequence.
- BERT4Rec [7]: It predicts the items masked in the sequence by utilizing the context around them.
- RLBL [17]: It is a circular log bilinear model that utilizes transition matrices to model the information of behavior types in a sequence.
- RIB [15]: It adds behavior embedding into the input layer and uses GRU and attention layers to capture the micro-behaviors in a sequence.
- BINN [16]: It uses a dual LSTM model to learn a user's long-term static interest and short-term dynamic interests and adds behavior information to LSTM.
- MSR [31]: It uses graph neural networks to capture relationships between items and uses GRU to model heterogeneous sequences at the same time.

Moreover, since our BMLP is based on MLP, we also include three closely related state-of-the-art MLP-based sequential recommendation methods (i.e., MLP-Mixer, MOI-Mixer, and FMLP-Rec).

- MLP-Mixer[5)] [26]: It is a new framework in computer vision that uses two independent MLPs to deal with classification tasks, which performs comparably to CNN and Transformer.
- MOI-Mixer [27]: It is improved based on MLP-Mixer [26] and proposes a multi-order sequential recommendation model.
- FMLP-Rec[6)] [28]: It introduces a filter to reduce the noise after encoding the items.

4.1.4 Experimental details

For our proposed model BMLP and all the baselines, we use grid search to select the best values of the parameters on the validation data. Specifically, we set the initial learning rate as 0.01 and the batch sample size as 512. The hidden layer dimension is selected from $\{64, 128, 256\}$. We fix the sequence length $L = 50$ and the auxiliary behavior subsequence length $L' = 5$. The number of heads and blocks are selected from $\{1, 2, 4, 8\}$ and $\{1, 2, 3\}$, respectively. To avoid overfitting, we choose the dropout rate from $\{0.2, 0.3, 0.4, 0.5\}$ and the regularization parameters from $\{0.0001, 0.001, 0.01, 0.1\}$. For Caser, we set the number of vertical and horizontal filters to 4 and 16, respectively, and choose the height of the horizontal filters from $\{2, 3, 4\}$. For RLBL, we select the window size from $\{1, 2, 3, 4, 5\}$. We use Adam for optimization. Finally, we evaluate the performance of our BMLP and all the baselines on the test data.

4.2 Overall Performance (RQ1)

In this section, we compare our model with the baselines on four datasets and show the results in Table 2. According to the experimental results, we can obtain the following observations.

- Our BMLP achieves a significant improvement compared with the baselines on all the four datasets, which clearly shows the effectiveness of our model. The MLP-based models can beat SASRec and GRU4Rec on RC15, Tmall, and ML1M. It demonstrates that a pure MLP model has the potential for sequential recommendation. Compared with MLP-based sequential recommendation methods, our BMLP shows a significant improvement in recommendation performance, which indicates that our model is able to capture the multi-behavioral dependencies more comprehensively.
- MSR and SRGNN are GNN-based models. MSR performs better than SRGNN on all the datasets. This shows that multi-behavior information contributes to the performance improvement. In contrast, BINN performs worse than GRU4Rec on some datasets, which means that BINN may not capture the dependencies between behaviors well.
- Caser is more prominent on Rec15, for which we speculate that users' short-term interests may dominate on Rec15. FMLP-Rec performs poorly on ML1M but well on the other datasets. It shows that although a dense data helps capture users' interests, it is also necessary to model them in a fine-grained manner. Otherwise, it may lose some important information using the filter layers.
- The graph-based methods SRGNN and MSR perform well

[5)] https://github.com/lucidrains/mlp-mixer-pytorch
[6)] https://github.com/Woeee/FMLP-Rec

Table 2: The overall performance of our BMLP and twelve baselines. Notice that the best results are marked in bold and the second best are underlined.

Dataset	Metric	Caser	GRU4Rec	SRGNN	BERT4Rec	SASRec	RLBL	RIB	BINN	MSR	MLP-Mixer	MOI-Mixer	FMLP	BMLP
Rec15	HR@10	0.461	0.437	0.483	0.389	0.437	0.331	0.387	0.468	0.605	0.478	0.482	0.541	**0.618**
	NDCG@10	0.247	0.249	0.285	0.196	0.248	0.160	0.213	0.257	0.378	0.259	0.260	0.255	**0.407**
	HR@20	0.612	0.561	0.608	0.535	0.566	0.459	0.521	0.611	0.715	0.635	0.640	0.672	**0.726**
	NDCG@20	0.287	0.281	0.317	0.233	0.283	0.191	0.248	0.293	0.408	0.298	0.299	0.286	**0.434**
Tmall	HR@10	0.098	0.352	0.274	0.168	0.369	0.243	0.269	0.298	0.298	0.292	0.317	0.381	**0.426**
	NDCG@10	0.076	0.239	0.187	0.096	0.281	0.163	0.188	0.199	0.195	0.203	0.215	0.309	**0.315**
	HR@20	0.122	0.411	0.319	0.225	0.415	0.281	0.306	0.349	0.348	0.361	0.369	0.421	**0.472**
	NDCG@20	0.081	0.252	0.196	0.109	0.292	0.181	0.202	0.213	0.211	0.221	0.228	0.315	**0.326**
ML1M	HR@10	0.215	0.268	0.249	0.145	0.243	0.205	0.229	0.273	0.265	0.264	0.269	0.116	**0.304**
	NDCG@10	0.119	0.152	0.141	0.072	0.132	0.102	0.122	0.155	0.150	0.145	0.151	0.058	**0.179**
	HR@20	0.311	0.361	0.343	0.237	0.345	0.302	0.335	0.376	0.373	0.367	0.378	0.169	**0.407**
	NDCG@20	0.142	0.180	0.167	0.095	0.159	0.129	0.150	0.181	0.176	0.171	0.179	0.068	**0.204**
UB	HR@10	0.126	0.171	0.142	0.098	0.181	0.093	0.132	0.141	0.172	0.145	0.144	0.158	**0.200**
	NDCG@10	0.086	0.104	0.083	0.055	0.106	0.055	0.080	0.083	0.099	0.085	0.085	0.103	**0.114**
	HR@20	0.152	0.219	0.186	0.138	0.238	0.122	0.168	0.189	0.225	0.187	0.187	0.192	**0.254**
	NDCG@20	0.085	0.116	0.092	0.063	0.121	0.062	0.089	0.097	0.117	0.099	0.099	0.113	**0.130**

on the Rec15 dataset, but not as well on the other datasets. The main reason is that the graph-based approach is more effective for datasets with short sequences. As shown in Table 1, the Rec15 dataset has the shortest user sequences.

4.3 Efficiency Analysis (RQ2)

In this subsection, we compare our model with two classical sequential recommendation models, i.e., GRU4Rec and SASRec, in terms of time complexity. For GRU4Rec, the time complexity of computing an item is $O(6d^2)$. Hence, for L items, the overall time complexity is $O(6Ld^2)$. SASRec has the following three operations: 1) mapping the hidden representations three times to obtain the query, key, and values; 2) calculating the attention scores, performing weighted sum; 3) passing through the fully connected layer. For the fully connected layer, the hidden unit is a constant independent of the input and output units, and the number of input unit is equal to the output unit. Therefore, their time complexity are $O(3Ld^2)$, $O(dL^2+Ld^2)$ and $O(2Ld)$, where L is the length of the sequence and d is the dimension of the representation. We can obtain the overall time complexity of SASRec as $O(4Ld^2+dL^2+2Ld)$. For our BMLP, the main modules are PIP and HIP, and their operations are fully connected layers. Therefore, their time complexities are $O(4dL)$ and $O(4mdL')$, where m denotes the number of auxiliary behaviors and L' denotes the length of the auxiliary behavior subsequence. We set the number of heads to 1 for brevity. Based on the above analysis, we can conclude that our BMLP is of a lower time complexity.

Experiments are conducted on Tmall with one Tesla V100 GPU and Intel(R) Xeon(R) Gold 6230R CPU @2.10GHz machine for all the compared models. Results of the the training time and performance metrics for our BMLP and baselines are shown in Table 3 and Table 4, respectively. We have the following observations:

Table 3: Results of training efficiency on Tmall.

Model	Time/Epoch	#Epoch	#Training Time
SASRec	22	10m	3h40m
GRU4Rec	13	12m	2h36h
Caser	15	8m	2h
SRGNN	11	17m	3h7m
BINN	12	22m	4h24m
BMLP	**30**	**3m**	**1h30m**

(1) Table 3 shows that our BMLP takes only three minutes for each epoch. It is much less than other models, leading to only 90 minutes for the whole training process. Eventually, our BMLP achieves around 3x and 2x speedup compared with BINN and SRGNN, respectively, demonstrating

Table 4: Efficiency and performance comparison on Tmall.

Model	#Training Time	#HR@10	#NDCG@10
SASRec	3h40m	0.362	0.277
GRU4Rec	2h36h	0.349	0.237
Caser	2h	0.094	0.073
SRGNN	3h7m	0.268	0.183
BINN	4h24m	0.293	0.196
BMLP	**1h30m**	**0.426**	**0.315**

the superiority of our BMLP in training efficiency.

(2) Table 4 shows the training time when the models converge and the performance of the corresponding models. We can see that our BMLP can achieve better performance with less time, which further demonstrates the higher efficiency of our BMLP in comparison with the other baseline models.

4.4 Study of Behavior Types (RQ3)

In this subsection, we mainly discuss how heterogeneous behaviors affect the recommendation performance of the model. We conduct four comparative experiments on each dataset.

- **S**: We ignore the behavior information, i.e., we do not take into account the specific behavior associated with each item.
- **B**: We introduce the behavior type by concatenating it with the item feature. Specifically, in Eq.(1), we only consider the behavior type information (represented as $B_{b_u^t}$) in the concatenation step.
- **T**: We introduce the behavior transition relationship. Specifically, we encode the behavior transition of two adjacent items, e.g., from purchase to click, and finally concatenate it with the item feature. In Eq.(1), we only consider the behavior transition relationship (represented as trans(b_u^t, b_u^{t+1}).
- **B+T (i.e., BMLP)**: It is a combination of **B** and **T**. Specifically, we add the features of the behavior types and behavior transition relationships as the overall behavior representations.

From the results in Fig 2, we can have the following observations: 1) If the behavior types are ignored, the performance of our model drops significantly. 2) The introduction of the behavior transition relationship has a greater performance improvement than the introduction of the behavior types. Intuitively, in a real e-commerce scenario, a user purchases an item, and later the user may click another in the next step.

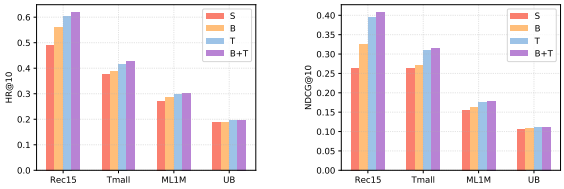


Fig. 2: The effectiveness of four different approaches of modeling the behavior types in our BMLP on four datasets. Notice that 'S', 'B', and 'T' denote modeling the sequences without the behavior types, with behavior types, and with behavior transition relationships, respectively.

Our behavior transition component can capture this micro transition from purchase to click. Therefore, the behavior transition relationships may better reflect the diversity and dependencies of the behaviors. 3) The results of our BMLP in **B+T** show that the combination of behavior types and behavior transition relations further improves the performance of our model. Therefore, we can see that the behavior types and the behavior transition relationships are complementary.

4.5 Ablation Study (RQ4)

To verify the contribution of each component to the overall performance, i.e., the effectiveness of SCB, FCB, and the impact of modeling auxiliary behavior subsequences or heterogeneous sequences, we conduct ablation studies.

- BMLP w/o FCB: We remove FCB. The variant of the model is insensitive to the feature information.
- BMLP w/o SCB: We remove SCB. The variant of this model does not capture the dependencies between items.
- BMLP w/o PIP: In order to verify the effectiveness of the PIP module, we use the HIP module alone as an ablation study. It does not tap changes in local interest.
- BMLP w/o HIP: We remove HIP to explore the performance impact of just using the subsequences of auxiliary behaviors.

The results of the above variants are shown in Table 5. We can obtain the following observations: 1) When any of the components are removed, the performance of the model degrades. It shows the effectiveness of each component. The performance drops more significantly after removing SCB comparing with that of removing FCB. This shows that SCB can capture the sequential patterns better, i.e., the sequen-

Table 5: Results of the ablation studies on four datasets.

	Rec15		Tmall		ML1M		UB	
	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
w/o FCB	0.581	0.406	0.398	0.301	0.277	0.160	0.175	0.099
w/o SCB	0.490	0.264	0.328	0.221	0.237	0.131	0.172	0.098
w/o PIP	0.567	0.325	0.385	0.283	**0.308**	**0.282**	0.168	0.093
w/o HIP	0.401	0.183	0.274	0.177	0.194	0.104	0.145	0.081
BMLP	**0.618**	**0.407**	**0.426**	**0.315**	0.304	0.179	**0.200**	**0.114**

tial information is relatively more important. 3) Firstly, the variant performs better when removing the PIP module on ML1M. Notice that the auxiliary and purchase behaviors do not exist in ML1M because it is a simulated data. So the PIP module is redundant to it. Secondly, we observe that the PIP module has a significant performance boost on the other datasets, with an increase of 6% on Rec15 and about 4% on Tmall and UB. Finally, according to the previous analysis of the results of Caser, we can see that short-term interest may dominate on Rec15. Intuitively, in real e-commerce scenarios, users often have some auxiliary behaviors about an item before purchasing it. Therefore, it can perceive this purchase intent sensitively as an auxiliary module. 4) The performance degradation is particularly significant when HIP is removed. This indicates that capturing the purchase intent through auxiliary behavior subsequences alone is insufficient. This also reflects that the sequential integrity is essential.

To further verify the ability of the SCB module to capture the sequential information, we design a comparison experiment by replacing the SCB module with a GRU module and a self-attention module.

Table 6: Results of comparison experiments on four data sets.

	Rec15		Tmall		ML1M		UB	
	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
BMLP_{GRU}	0.454	0.258	0.396	0.288	0.281	0.162	0.189	0.107
BMLP_{Att}	0.523	0.296	0.419	0.311	0.293	0.169	**0.209**	**0.118**
BMLP_{MLP}	**0.618**	**0.407**	**0.426**	**0.315**	**0.304**	**0.179**	0.200	0.114

The results of the comparison experiments are presented in Table 6, and the key observations are summarized below. Firstly, after replacing the SCB module with the GRU or the self-attention module, our BMLP(MLP) still outperforms BMP(GRU) and BMLP(Att) on three datasets. This demonstrates the effectiveness of the SCB module in capturing the sequential information. Secondly, by integrating the GRU or the self-attention module into our BMLP, it achieves a significant improvement in overall performance compared with GRU4Rec or SASRec. This highlights the versatility and generality of our proposed architecture.

4.6 Purchase Intent Analysis (RQ5)

To investigate how the PIP module contributes to the final purchased item prediction, we conduct an empirical study on the behavior types of the predicted items. Specifically, we select the last purchased item of each sequence as the test set in the previous experiments. In contrast, we choose an auxiliary behavior between the last two purchase behaviors as the test set in this study. To ensure fairness of the comparison experiment, we strictly maintain the integrity of the sequence. ML1M is a simulated dataset without purchase and auxiliary behaviors. Therefore, we do not include ML1M in the study. From Figure 3, we can see that the accuracy of predicting purchase behavior is higher than that of predicting auxiliary behavior. The PIP module can capture the user's purchase intent well, leading to performance improvement.

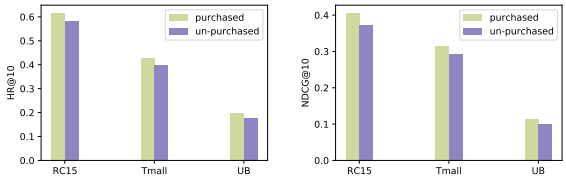


Fig. 3: Recommendation performance in predicting purchased items versus un-purchased items on Rec15, Tmall, and UB. Notice that the behaviors in ML1M are simulated, which are thus not included.

4.7 Performance Comparison with Heterogeneous Non-sequential Recommendation Methods (RQ6)

In this subsection, we compare several classic heterogeneous non-sequential recommendation methods and conduct a deep analysis of how the appearance or non-appearance of a target item in a historical sequence affects the performance. Firstly, we conduct an analysis on the percentage of target items appearing in recent historical behaviors on all datasets. Among

Table 7: The results of the grouped experiments on Tmall and UB. The term "examined" refers to the results of the test samples where the target item appears in the previous heterogeneous sequence, while "unexamined" indicates the contrary. The term "average" represents the overall results without distinguishing whether the target item appears in previous heterogeneous sequences.

	Tmall						UB					
	Examined		Unexamined		Average		Examined		Unexamined		Average	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
MBGMN	0.0311	0.0185	0.0278	0.0168	0.0298	0.0178	0.0171	0.0079	0.0152	0.0066	0.0162	0.0073
CML	0.0322	0.0199	0.0307	0.0191	0.0316	0.0196	0.0186	0.0092	0.0175	0.0086	0.0181	0.0089
MMCLR	0.0443	0.0268	0.0392	0.0233	0.0423	0.0254	0.0261	0.0120	0.0211	0.0103	0.0237	0.0112
SASRec	0.5575	0.4373	0.0858	0.0459	0.3688	0.2812	0.2931	0.1737	0.0617	0.0332	0.1813	0.1058
BMLP	**0.6483**	**0.4889**	**0.0916**	**0.0523**	**0.4263**	**0.3148**	**0.3235**	**0.1844**	**0.0687**	**0.0391**	**0.2004**	**0.1142**

all the test samples, the percentage of the target items appearing in the previous historical behaviors is 60.1%, 51.7%, 0%, and 99.6% on Tmall, UB, ML1M, and Rec15, respectively. The ML1M dataset is a simulated dataset that lacks purchase and click behaviors. In contrast, the Rec15 dataset comprises a significant number of instances where items are first clicked and subsequently purchased, leading to a remarkably high occurrence rate of 99.6%. To ensure the rigorousness of our experiments, we divide the test data into two groups, i.e., an unexamined test set and an examined test set on Tmall and UB. Notice that "examined" means that the target item appears in a previous heterogeneous sequence, while "unexamined" denotes the contrary. We conduct an evaluation of our proposed BMLP, in comparison with three heterogeneous non-sequential recommendation methods, i.e., MBGMN [12], CML [36] and MMCLR [37] and a classical sequential recommendation method SASRec. The results are shown in Table 7. According to the experimental results, we can obtain the following observations: 1) Two sequential recommendation methods demonstrate a slightly superior performance on the examined test set compared to the overall average, while they experience a substantial decline on the unexamined test set. It suggests that the presence of the target item in the historical sequences simplifies the prediction task to some extent. 2) Regarding non-sequential recommendation, the performance degradation on the unexamined test set is not that much, indicating that the non-sequential recommendation methods are not particularly sensitive to the occurrence of the target item in historical sequences. 3) When comparing the results on the unexamined test set, our proposed method BMLP, still outperforms both the heterogeneous non-sequential recommendation methods and the sequential recommendation method SASRec. Notice that a previous study [35] has also observed that heterogeneous non-sequential recommendation methods exhibit particularly poor performance.

4.8 Hyperparameter Sensitivity (RQ7)

In this subsection, we explore the impact of two hyperparameters on our model, i.e., the number of heads and the lengths of the most recent auxiliary behavior subsequences.

4.8.1 The impact of the number of heads

We conduct the corresponding experiments with $H \in \{1, 2, 4, 8\}$. The experimental results on four datasets are shown in Fig 4. We find that the overall performance of our model reaches the best when $H = 2$. When H is too large or too small, the performance is slightly worse. A larger value of H means that the feature dimension on each head is correspondingly smaller. If the feature dimension is too small, the information of the subspace cannot be accurately represented. A moderate value of H can balance the dimensionality of each subspace and the number of subspaces.

4.8.2 The impact of the length of the auxiliary behavior subsequences

The recent auxiliary behavior subsequences are used as the input of the PIP module, of which the length is important

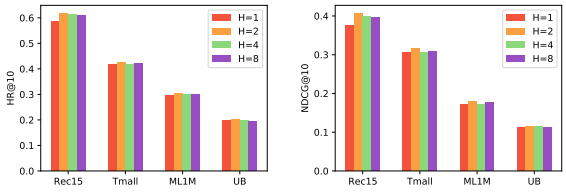
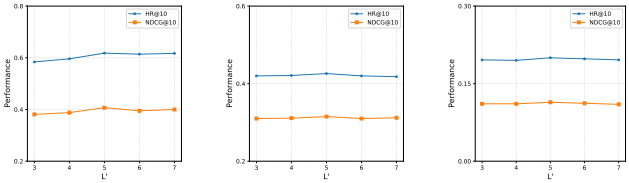


Fig. 4: Recommendation performance of our BMLP with different numbers of heads on four datasets.



(a) Rec15 (b) Tmall (c) UB

Fig. 5: Recommendation performance of our BMLP with different lengths of the most recent auxiliary behavior subsequences, i.e., $L' \in \{3, 4, 5, 6, 7\}$, on four datasets.

for the dissection of the PIP module. Therefore, we conduct comparison experiments with different lengths of these subsequences on each dataset, and the values of the length are selected from $\{3, 4, 5, 6, 7\}$. The experimental results are shown in Figure 5. We can see that the overall performance of our model reaches the best when $L' = 5$. When the lengths of the most recent auxiliary behavior subsequences are relatively large or small, the model performance will decrease. The previous auxiliary behaviors become less important for the next purchased item when L' is too large. Moreover, when the length is too small, it may introduce noise to the short-term purchase interest learned by the PIP module.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel pure MLP-based solution, i.e., behavior-aware MLP (BMLP), for heterogeneous sequential recommendation (HSR). Specifically, our BMLP contains three modules, i.e., heterogeneous interest perception (HIP), purchase intent perception (PIP), and recommendation. The main structure of HIP and PIP consists of two MLPs, one dealing with sequential information and the other with feature representation. In the HIP module, we consider behavior types and behavior transition relations for modeling behaviors at multiple granularities to capture a user's heterogeneous interests. In the PIP module, we adaptively aggregate the recent auxiliary behavior subsequences to obtain a user's dynamic purchase intent. Finally, the recommendation module fuses the heterogeneous interest and the purchase intent for prediction. Extensive experiments show that our BMLP outperforms the state-of-the-art baselines.

For future works, we are interested in leveraging rich contextual information, such as review comments and item attributes. Moreover, we intend to generalize our model to multi-task scenarios in order to improve the performance on both the purchase and non-purchase behaviors.

Acknowledgements We thank the support of National Natural Science Foundation of China No. 62172283 and No. 62272315. We thank Miss Qianzhen Rao for her helpful discussions.

References

1. Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.
2. Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *Proceedings of the 11th ACM Conference on Recommender Systems*, pages 130–137, 2017.
3. Jiaxi Tang and Ke Wang. Personalized top-N sequential recommendation via convolutional sequence embedding. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, pages 565–573, 2018.
4. Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. A simple convolutional generative network for next item recommendation. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, pages 582–590, 2019.
5. Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *Proceedings of 2018 IEEE International Conference on Data Mining*, pages 197–206, 2018.
6. Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. Deep session interest network for click-through rate prediction. *arXiv preprint arXiv:1905.06482*, 2019.
7. Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th*

ACM International Conference on Information and Knowledge Management, pages 1441–1450, 2019.

8. Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *Proceedings of the 33th AAAI Conference on Artificial Intelligence*, number 01, pages 346–353, 2019.
9. Chen Ma, Liheng Ma, Yingxue Zhang, Jianing Sun, Xue Liu, and Mark Coates. Memory augmented graph neural networks for sequential recommendation. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, number 04, pages 5045–5052, 2020.
10. Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. Graph contextualized self-attention network for session-based recommendation. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 3940–3946, 2019.
11. Yitong Pang, Lingfei Wu, Qi Shen, Yiming Zhang, Zhihua Wei, Fangli Xu, Ethan Chang, Bo Long, and Jian Pei. Heterogeneous global graph neural networks for personalized session-based recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, pages 775–783, 2022.
12. Lianghao Xia, Yong Xu, Chao Huang, Peng Dai, and Liefeng Bo. Graph meta network for multi-behavior recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 757–766, 2021.
13. Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. Multi-behavior recommendation with graph convolutional networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 659–668, 2020.
14. Wanqi Ma, Xiancong Chen, Weike Pan, and Zhong Ming. VAE++: Variational autoencoder for heterogeneous cne-class collaborative filtering. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, pages 666–674, 2022.
15. Meizi Zhou, Zhuoye Ding, Jiliang Tang, and Dawei Yin. Micro behaviors: A new perspective in e-commerce recommender systems. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, pages 727–735, 2018.
16. Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1734–1743, 2018.
17. Qiang Liu, Shu Wu, and Liang Wang. Multi-behavioral sequential prediction with recurrent log-bilinear model. *IEEE Transactions on Knowledge and Data Engineering*, 29(6):1254–1267, 2017.
18. Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, Lixin Zou, Yiding Liu, and Dawei Yin. Deep multifaceted transformers for multi-objective ranking in large-scale e-commerce recommender systems. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pages 2493–2500, 2020.
19. Wen Wang, Wei Zhang, Shukai Liu, Qi Liu, Bo Zhang, Leyu Lin, and Hongyuan Zha. Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction. In *Proceedings of The Web Conference 2020*, pages 3056–3062, 2020.
20. Fuyu Lv, Taiwei Jin, Changlong Yu, Fei Sun, Quan Lin, Keping Yang, and Wilfred Ng. SDM: Sequential deep matching model for online large-scale recommender system. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2635–2643, 2019.
21. Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 811–820, 2010.
22. Ruining He and Julian McAuley. Fusing similarity models with markov chains for sparse sequential recommendation. In *Proceedings of 2016 IEEE International Conference on Data Mining*, pages 191–200, 2016.
23. Ruining He, Wang-Cheng Kang, and Julian McAuley. Translation-based recommendation. In *Proceedings of the 11th ACM Conference on Recommender Systems*, pages 161–169, 2017.
24. Zhe Zhao, Zhiyuan Cheng, Lichan Hong, and Ed H Chi. Improving user topic interest profiles by behavior factorization. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1406–1416, 2015.
25. Zhuo-Xin Zhan, Ming-Kai He, Wei-Ke Pan, and Zhong Ming. TransRec++: Translation-based sequential recommendation with heterogeneous feedback. *Frontiers of Computer Science*, 16(2):1–3, 2022.
26. Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. MLP-Mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34, 2021.
27. Hojoon Lee, Dongyoon Hwang, Sunghwan Hong, Changyeon Kim, Seungryong Kim, and Jaegul Choo. Moi-mixer: Improving mlp-mixer with multi order interactions in sequential recommendation. *arXiv preprint arXiv:2108.07505*, 2021.
28. Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. Filter-enhanced mlp is all you need for sequential recommendation. In *Proceedings of the International Conference on World Wide Web*, pages 2388–2399, 2022.
29. Muyang Li, Xiangyu Zhao, Chuan Lyu, Minghao Zhao, Runze Wu, and Ruocheng Guo. Mlp4rec: A pure mlp architecture for sequential recommendations. *arXiv preprint arXiv:2204.11510*, 2022.
30. Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, 28:802–810, 2015.
31. Wenjing Meng, Deqing Yang, and Yanghua Xiao. Incorporating

user micro-behaviors and item knowledge into multi-task learning for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1091–1100, 2020.

32. Chong Chen, Min Zhang, Yongfeng Zhang, Weizhi Ma, Yiqun Liu, Shaoping Ma. Efficient heterogeneous collaborative filtering without negative sampling for recommendation. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pages 19–26, 2020.
33. Mingshi Yan, Zhiyong Cheng, Chen Gao, Jing Sun, Fan Liu, Fuming Sun, and Haojie Li. Cascading residual graph convolutional network for multi-behavior recommendation. *arXiv preprint arXiv:2302.03525, 2022.*
34. *Wang, Yuhao and Lam, Ha Tsz and Wong, Yi and Liu, Ziru and Zhao, Xiangyu and Wang, Yichao and Chen, Bo and Guo, Huifeng and Tang, Ruiming. Multi-task deep recommender systems: A survey.* arXiv preprint arXiv:2205.13128, 2023.
35. *Cho, Junsu and Hyun, Dongmin and won Lim, Dong and jae Cheon, Hyeon and Park, Hyoung-iel and Yu, Hwanjo. Dynamic multi-behavior sequence modeling for next item recommendation. In* Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4199–4207, 2023.*
36. *Wei Wei, Chao Huang, Lianghao Xia, Yong Xu , Jiashu Zhao, and Dawei Yin. Contrastive meta learning with behavior multiplicity for recommendation. In* Proceedings of the 15th ACM International Conference on Web Search and Data Mining, pages 1120–1128, 2022.
37. Yiqing Wu, Ruobing Xie, Yongchun Zhu, Xiang Ao, Xin Chen, Xu Zhang, Fuzhen Zhuang, Leyu Lin, and Qing He. Multi-view multi-behavior contrastive learning in recommendation. In *Proceedings of the 27th International Conference on Database Systems for Advanced Application*, pages 166–182, 2022.
38. Zhicheng Qiao, Hui Yan, and Lei Han. Mixmbr: Contrastive learning for multi-behavior recommendation. In *Proceedings of the 28th International Conference on Database Systems for Advanced Applications*, pages 434–445, 2023.