Dual-Task Learning for Multi-Behavior Sequential Recommendation

Jinwei Luo¹,²,³, Mingkai He¹,²,³, Xiaolin Lin¹,²,³, Weike Pan¹,²,³*, Zhong Ming¹,²,³*

luojinwei2016@email.szu.edu.cn, 1910273003@email.szu.edu.cn, linxiaolin2021@email.szu.edu.cn, {panweike, mingz}@szu.edu.cn

¹ College of Computer Science and Software Engineering, Shenzhen University, China

² Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University, China

³ National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, China
Problem Definition

Multi-Behavior Sequential Recommendation (MBSR)

- **Input**: An interaction sequence of (item, behavior) pairs
  \[ S_u = \{ (i_{u1}^1, b_{u1}^1), \ldots, (i_{u\ell}^\ell, b_{u\ell}^\ell), \ldots, (i_{uL}^L, b_{uL}^L) \} \], where \( i_{u\ell}^\ell \in I \) denotes the \( \ell \)th item interacted by user \( u \) with behavior \( b_{u\ell}^\ell \in B \).

- **Goal**: Predict the next likely-to-purchase **new item** \( i \) of a user \( u \) from \( I \setminus I_u \) at timestamp \( L + 1 \). Notice that \( I_u \) denotes the set of items interacted by user \( u \).
Motivation

- MBSR that exploits users’ heterogeneous interactions in sequences has received relatively little attention.
- Existing works often overlook the complementary effect of different perspectives when addressing the MBSR problem.
  - Some works take a multi-behavior interaction sequence as a behavior-agnostic item sequence and a behavior sequence.
  - Some works take a multi-behavior interaction sequence as some behavior-specific item sub-sequences.
Challenges

- **Heterogeneity of a user’s intention and the context information.** Users often have different behavioral intentions (i.e., examination, purchase) at different time steps, and there are different contextual information of behaviors in the historical process.

- **Sparsity of the interactions of target behavior.** A user’s purchase behavior is often much sparser in his or her interaction history, hindering us to learn a user’s purchase preferences effectively.
To release the potential of multi-behavior interaction sequences, we propose a novel framework named NextIP that adopts a dual-task learning strategy to convert the problem to two specific tasks, i.e., next-item prediction and purchase prediction.

For next-item prediction, we design a target-behavior aware context aggregator (TBCG), which utilizes the next behavior to guide all kinds of behavior-specific item sub-sequences to jointly predict the next item.

For purchase prediction, we design a behavior-aware self-attention (BSA) mechanism to extract a user’s behavior-specific interests (i.e., virtual items) in a certain time interval and treat them as negative samples to learn the user’s purchase preferences.
Single-Behavior Sequential Recommendation (SBSR)

- The earliest works of sequential recommendation are based on Markov chains (MCs) and matrix factorization (MF) models.
- With the development of deep learning, various techniques are applied to sequential recommendation.
- Based on the success of Transformer and the attention mechanism in NLP [Vaswani et al., 2017], SASRec [Kang and McAuley, 2018] is proposed to use some self-attention modules to encode the sequential information of historical sequences.
- In addition, many researchers propose to leverage graph neural network (GNN) to model the sequential data, especially in session-based recommendation.
Multi-Behavior Recommendation (MBR)

**MB-GMN** [Xia et al., 2021] further proposes a graph meta network to capture different behavior semantics and relations, which is found to be very effective in MBR.

**VAE++** [Ma et al., 2022] is the most recent variational autoencoder (VAE)-based method that simultaneously models three types of signals, including the target behavior, the auxiliary behavior and their mixed behaviors with multiple encoders and one single decoder.

Notice that there are some significant differences between MBR and MBSR, because capturing the complex sequential information (e.g., global/local preferences, behavior transitions, behavior tendency/continuity, long/short sequences, periodicity, etc.) is not a trivial task.
**Multi-Behavior Sequential Recommendation (MBSR)**

- The first category is to model a multi-behavior interaction sequence as a behavior-agnostic item sequence and a corresponding behavior sequence. [Li et al., 2018, Meng et al., 2020].
- The advantage of this type of methods is that they allow the models to distinguish different types of behaviors while retaining the integrity of an entire interaction sequence.

- The second category is to divide a multi-behavior interaction sequence into some behavior-specific item sub-sequences and model them jointly [Yu et al., 2022, Xie et al., 2020].
- The merit of this modeling perspective is that the problem can be simplified into modeling the item sequences since the semantics of each behavior type can be learned via different parameters.
## Table: Some notations and their explanations used in the paper.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{U} = { u } )</td>
<td>the whole set of users</td>
</tr>
<tr>
<td>( \mathcal{I} = { i } )</td>
<td>the whole set of items</td>
</tr>
<tr>
<td>( \mathcal{B} )</td>
<td>the set of behaviors</td>
</tr>
<tr>
<td>( i_u^\ell \in \mathcal{I} )</td>
<td>the ( \ell )th item interacted by user ( u )</td>
</tr>
<tr>
<td>( b_u^\ell \in \mathcal{B} )</td>
<td>the ( \ell )th behavior type interacted by user ( u )</td>
</tr>
<tr>
<td>( S_u = {(i_u^\ell, b_u^\ell)} )</td>
<td>the interaction sequence of user ( u )</td>
</tr>
<tr>
<td>( d \in \mathbb{R} )</td>
<td>number of latent dimensions</td>
</tr>
<tr>
<td>( L \in \mathbb{R} )</td>
<td>length of the sequence</td>
</tr>
<tr>
<td>( U \in \mathbb{R}^{\mathcal{U} \times d} )</td>
<td>user embedding matrix</td>
</tr>
<tr>
<td>( u_u \in \mathbb{R}^{1 \times d} )</td>
<td>user embedding of user ( u )</td>
</tr>
<tr>
<td>( H \in \mathbb{R}^{\mathcal{I} \times d} )</td>
<td>item embedding matrix</td>
</tr>
<tr>
<td>( h_i \in \mathbb{R}^{1 \times d} )</td>
<td>item embedding of item ( i )</td>
</tr>
<tr>
<td>( B \in \mathbb{R}^{\mathcal{B} \times d} )</td>
<td>behavior embedding matrix</td>
</tr>
<tr>
<td>( b_{b_u^\ell} )</td>
<td>behavior embedding of behavior ( b_u^\ell )</td>
</tr>
<tr>
<td>( P \in \mathbb{R}^{L \times d} )</td>
<td>position embedding matrix</td>
</tr>
</tbody>
</table>
### Table: Some notations and their explanations used in the paper (cont.).

<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$SAB(\cdot)$</td>
<td>self-attention block</td>
</tr>
<tr>
<td>$X_u^{(k)} \in \mathbb{R}^{L \times d}$</td>
<td>representation matrix of a behavior-agnostic item sequence in $k$th layer of SAB</td>
</tr>
<tr>
<td>$X_u^{e(k)} \in \mathbb{R}^{L \times d}$</td>
<td>representation matrix of an examination-specific item sequence in $k$th layer of SAB</td>
</tr>
<tr>
<td>$x_{u,\ell} \in \mathbb{R}^{1 \times d}$</td>
<td>behavior-agnostic sequence representation of user $u$ at step $\ell$</td>
</tr>
<tr>
<td>$g'_{u,\ell} \in \mathbb{R}^{1 \times d}$</td>
<td>behavior-aware context representation of user $u$ at step $\ell$</td>
</tr>
</tbody>
</table>
Luo et al., (Shenzhen University)

**NextIP (1/3)**

**Proposed Method**

**Figure**: The overview of our proposed NextIP.
In general, our NextIP takes a multi-behavior interaction sequence of user $u$ as input and obtains the representations of the behavior sequence, each behavior-specific item sub-sequence and the behavior-agnostic item sequence as well as the user embedding in the embedding layer.

These representations are then fed into two well-designed tasks, i.e., a next-item prediction task (in the bottom left corner) and a purchase prediction task (in the bottom right corner) to capture both the user’s behavior-aware short-term interests and purchase-oriented long-term preferences, which are finally used to fulfill the final prediction.
NextIP (3/3)

- As a response to the aforementioned two challenges, we propose a novel and effective framework named NextIP that adopts a dual-task learning strategy to convert the problem to two specific tasks, i.e., next-item prediction and purchase prediction.

- We address the heterogeneity challenge in task 1 (in the bottom left corner of Figure 1) by designing a novel target-behavior aware context aggregator (TBCG) to transfer the unique knowledge of different types of behaviors so as to predict the next item in a behavior-aware manner.

- We address the sparsity challenge in task 2 (in the bottom right corner of Figure 1). Specifically, all the items that a user has interacted with are treated as positive samples in task 1, while in task 2, we propose to treat the items associated with auxiliary behaviors as negative signals to refine the learning of a user’s purchase-oriented preferences.
We present an embedding layer to convert an input multi-behavior interaction sequence $S_u$ to different embeddings.

Through embedding look-up operations from $H$, we can retrieve and stack the item embeddings of $S_u$ as an embedding matrix
\[
E = \begin{bmatrix}
h_{i_1}^1; \ldots; h_{i_\ell}^\ell; \ldots; h_{i_L}^L
\end{bmatrix} \in \mathbb{R}^{L \times d}.
\]

Following [Kang and McAuley, 2018], the representation of a behavior-agnostic item sequence can be obtained by adding two embedding matrices, i.e.,
\[
X_u^{(0)} = E + P.
\]
In order to unlock the potential of a multi-behavior interaction sequence, we further represent a user’s behavior-specific item sequence by behavior-specific masking.

We take the examination-specific item sequence as an example.

\[
X_u^{e(0)} = X_u^{(0)} \otimes M^e,
\]

where \( \otimes \) is the element-wise product.

\[ M^e = [m_1^e; \ldots; m_{\ell}^e; \ldots; m_L^e] \in \mathbb{R}^{L \times d} \] denotes the examination-specific mask matrix, in which \( m_{\ell}^e \) is a vector of ones if \( b_{u\ell} = e \), and a vector of zeros otherwise.

Similarly, we can obtain \( X_u^{f(0)} \), \( X_u^{c(0)} \) and \( X_u^{p(0)} \) as the embedding matrices of the other three behavior-specific item sequences. Note that \( f \), \( c \) and \( p \) denote add-to-favorite, add-to-cart and purchase, respectively.
Task 1: Next-Item Prediction

![Diagram of Next-Item Prediction Algorithm]

- **Item Sequence Encoder**
  - $X_u^{(0)}$
  - $X_u^{(0)}$
  - $X_u^{(0)}$
  - $X_u^{(0)}$

- **Target- Behavior Aware Context Aggregator (TBCG)**
  - $b_{u+1}$
  - $h_i$
  - $r_{t+1,i}$

- **Sequential Padding**
  - $X_u^e$
  - $X_u^l$
  - $X_u^p$

- **Context Aggregation**
  - $x_{u,e}$
  - $x_{u,l}$
  - $x_{u,p}$

- **Mean Pooling**
  - $g_{u,e}$
  - $g'_{u,l}$

- **Candidate Item**
  - $g'_{u,l} + x_{u,l}$

**Notes:**
- Luo et al., (Shenzhen University)
- NextIP
- CIKM 2022
To obtain different context information in all the behavior-agnostic and behavior-specific sequences, we choose self-attention block (SAB) [Kang and McAuley, 2018] for the item sequence encoder since it is known effective.

The parameters of each self-attention network are independent across different kinds of item sequences, which are used to capture different semantic information and transition patterns.

For instance, the transitions in an examination-specific item sequence usually indicate similarity among the items while those in a purchase-specific item sequence often mean complementarity to some extent.
Proposed Method

Task 1: Next-Item Prediction

Item Sequence Encoder (2/3)

Omitting the formulas of the layer normalization and residual connection, each SAB consists of a self-attention layer (SAL), and a feed-forward layer (FFL),

\[ SAB(X) = FFL(SAL(X)), \]

\[ X' = SAL(X) = (softmax(\frac{QK^T}{\sqrt{d}}) \otimes \Delta)V, \]

\[ FFL(X') = ReLU(X'W_1 + b_1)W_2 + b_2, \]

where \( Q = XW_Q, \ K = XW_K \) and \( V = XW_V \) with \( W_Q, W_K, W_V \in \mathbb{R}^{d \times d} \) are the projected query, key and value matrices, respectively. Note that \( \sqrt{d} \) in SAL(\cdot) is the scaling factor to prevent overlarge values of the inner product and \( \Delta \) is the causality mask used to ensure that only the previous \( \ell \) items are taken into account when predicting the \((\ell + 1)\)th item. \( W_1, W_2 \in \mathbb{R}^{d \times d} \) and \( b_1, b_2 \in \mathbb{R}^{1 \times d} \) are learnable weights and biases of the two-layer network, respectively.
By feeding $X_u^{(0)}$ into the corresponding $K$ layers of SABs, we can obtain $X_u^{(K)} = [x_{u,1}^{(K)}; \ldots; x_{u,\ell}^{(K)}; \ldots; x_{u,L}^{(K)}]$, where $x_{u,\ell}^{(K)} \in \mathbb{R}^{1 \times d}$ denotes the representation of the user’s behavior-agnostic context at step $\ell$.

Similarly, we can obtain $X_u^{e(K)}$, $X_u^{f(K)}$, $X_u^{c(K)}$, $X_u^{p(K)}$ for the behavior-specific item sequences. Note that we remove the superscript $K$ for brevity in the figure and in the subsequent text.
Different types of behavior-specific item sequences can provide different contextual information, which thus should be explicitly exploited.

For instance, the records of a user’s recent examinations on some earphones are useful signals that the next purchase might be an earphone, and knowing a user bought an iPhone last month can inform us to predict that the user is likely to buy an AirPods.

However, the context information needs to be well balanced when predicting different behaviors, since the dependency across different behaviors is complex.
Therefore, we design a target-behavior aware context aggregator (TBCG), which takes a user’s target behavior embedding to improve the context aggregation.

Specifically, at the current step $\ell$, we stack up the context representations of different behaviors (i.e., $[x^e_{u,\ell}; x^f_{u,\ell}; x^c_{u,\ell}; x^p_{u,\ell}]$) and take them as keys and values, while leveraging the user’s behavior embedding at the next step $b^l_{b^l+1}$ as a query to adaptively extract the context that is more relevant to predict the next item,
Target-Behavior Aware Context Aggregator (3/4)

\[ Q'_\ell = b_{u+1} W_{Q'}, \]
\[ K'_\ell = \begin{bmatrix} x_{e,u,\ell}; x_{f,u,\ell}; x_{c,u,\ell}; x_{p,u,\ell} \end{bmatrix} W_{K'}, \]
\[ V'_\ell = \begin{bmatrix} x_{e,u,\ell}; x_{f,u,\ell}; x_{c,u,\ell}; x_{p,u,\ell} \end{bmatrix} W_{V'}, \]
\[ g_{u,\ell} = \text{softmax}\left( \frac{Q'_\ell K'_\ell^T}{\sqrt{d}} \right) V'_\ell, \]

where \( Q'_\ell \in \mathbb{R}^{1 \times d} \), \( K'_\ell \in \mathbb{R}^{4 \times d} \) and \( V'_\ell \in \mathbb{R}^{4 \times d} \) are the projected query, key and value matrices, respectively. And \( W_{Q'}, W_{K'}, W_{V'} \in \mathbb{R}^{d \times d} \) are the learnable weight matrices. Note that \( g_{u,\ell} \) denotes the gated context representation of user \( u \) at step \( \ell \) considering the user’s target behavior.
Target-Behavior Aware Context Aggregator (4/4)

We further obtain the refined behavior-aware context representation by combining $\mathbf{g}_{u,\ell}$ with the context representation of user $u$ at step $\ell$ for behavior type $\ast$, i.e., $\mathbf{x}_{u,\ell}^\ast$, via mean pooling,

$$
\mathbf{g}'_{u,\ell} = \frac{\mathbf{g}_{u,\ell} + \mathbf{x}_{u,\ell}^\ast}{2},
$$

(9)

where $\ast$ denotes the next behavior of the current step, i.e., $b_{u,\ell+1} \in \mathcal{B}$.

The main idea behind Eq.(9) is to enhance the contribution of the context representation of the next behavior type at the current step in the training phase. Note that in the evaluation phase, we will fix the next behavior type as purchase $p$, since our goal is next purchased item prediction.
Loss Function

- We predict the probability that user $u$ will interact with item $i$ at the $(\ell + 1)$th step as follows,

$$\hat{r}_{\ell+1,i} = (x_{u,\ell} + g'_{u,\ell})(h_i)^T.$$  \hspace{1cm} (10)

- We adopt the typical binary cross-entropy loss for task 1,

$$L_1 = - \sum_{u \in U} \sum_{\ell = 2}^{L+1} \delta(i_{u,\ell}^\ell)[\log(\sigma(\hat{r}_{\ell,i_u^\ell})) + \log(1 - \sigma(\hat{r}_{\ell,j}^\ell))],$$  \hspace{1cm} (11)

where $j$ is a negative item randomly sampled from $I \setminus I_u$ for each position $\ell \in \{2, ..., L + 1\}$. The indicator function $\delta(i_{u}^\ell) = 1$ if $i_{u}^\ell$ is not a padding item, and 0 otherwise. The indicator function $\delta(i_{u}^\ell)$ is used to ignore the loss value when $i_{u}^\ell$ is a padding item, since we follow the common practice and use a padding item to pad the user sequence to a same length $L$. 

Luo et al., (Shenzhen University)  
NextIP  
CIKM 2022
Task 2: Purchase Prediction

Behavior-Aware Self-Attention (BSA)

Virtual Items’ Representations

Contrastive Loss

User Embedding

\( E \)

\( N_u \)

\( n_{u,t_1} \)

\( n_{u,t_2} \)

\( n_{u,t_3} \)

\( h_y \)

+ positive sample

- negative sample

Luo et al., (Shenzhen University)
Task 1 is designed to predict the next item of all behaviors and to learn a user’s short-term interests while task 2 aims to learn a user’s purchase-oriented long-term preferences.

In addition, a user’s purchase behavior is often much sparser in his or her interaction history, hindering us to learn the user’s purchase preferences effectively.

To address the above issues, we leverage a user’s behavior transitions to learn his or her purchase-oriented preferences in a self-supervised manner. For instance, users often interact with different brands of earphones for comparison via some auxiliary behaviors (e.g., examination) before making a purchase decision.

Therefore, we should pay attention to distinguishing the purchased item from the previous items with some auxiliary behaviors.
The intentions reflected in a user’s interaction history are usually diverse and uncertain.

For example, a user might examine clothes before purchasing an earphone, and thus assuming the user prefers the earphone to each cloth may not be reasonable.

To alleviate this uncertainty issue, we design a behavior-aware self-attention (BSA) mechanism to aggregate the collection of the some recently interacted items of each kind of behavior into a virtual item w.r.t. each step.
**Behavior-Aware Self-Attention (3/4)**

Figure: An example of the attention mask in our NextIP with $L = 6$ and $L_{\text{clip}} = 3$.

where $M \in \mathbb{R}^{L \times L}$ denotes the behavior-aware attention mask, $M_{\ell, \ell'} = 1$ if $b_u^\ell = b_u^{\ell'}$ and $\ell - \ell' \leq L_{\text{clip}}$, and $M_{\ell, \ell'} = 0$ otherwise. The usage of a length clip is to ensure that the aggregated items correspond to a similar intention of user $u$. 
We use self-attention to aggregate each behavior-specific item set into a virtual item representation,

\[ N_u = (\text{softmax}(\frac{Q''K''^T}{\sqrt{d}}) \otimes A)V'', \] (12)

where \( Q'' = EW_{Q''}, K'' = EW_{K''} \) and \( V'' = EW_{V''} \) with \( W_{Q''}, W_{K''}, W_{V''} \in \mathbb{R}^{d \times d} \) as the projected query, key and value matrices, respectively.

After BSA, for user \( u \), we can obtain \( N_u = [n_{u,1}; \ldots; n_{u,\ell}; \ldots; n_{u,L}] \), where \( n_{u,\ell} \in \mathbb{R}^{1 \times d} \) denotes the representation of a virtual item that user \( u \) has recently interacted with using a certain behavior at step \( \ell \).
Loss Function

- We treat the virtual items’ representations of every auxiliary behavior (e.g., examination) before a purchased item as negative samples, which will be used to learn the user’s purchase-oriented preferences.

\[
g(h_y) = \exp \left( u_u W h_y^T / \rho \right),
\]

\[
g(n_{u,t}) = \exp \left( u_u W n_{u,t}^T / \rho \right),
\]

\[
L_2 = -\sum_{u \in U} \sum_{y \in \mathcal{Y}_u} \log \frac{g(h_y)}{g(h_y) + \sum_{t \in \mathcal{T}(u,y)} g(n_{u,t})},
\]

where \( y \in \mathcal{Y}_u \) is an item purchased by user \( u \), \( u_u \in \mathbb{R}^{1 \times d} \) denotes the user embedding, and \( \rho \) is the temperature parameter. \( W \) is a learnable weight matrix to perform feature transformation, and \( \mathcal{T}(u,y) \) is a set containing the step indices of each type of auxiliary behavior that is closest to the user’s purchased item \( y \).
Learning and Final Prediction (1/2)

Task 1
\[ g'_{u,\ell} + x_{u,\ell} \] 

Task 2
\[ u_u \]

Final Prediction
\[ z_{\ell} \]

Candidate Item
\[ h_i \]

Prediction
\[ \hat{\delta}_{\ell+1,i} \]
We train our NextIP in an end-to-end fashion by minimizing the loss on both task 1 and task 2,

\[ \mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2. \]  

(16)

Note that simply setting the weight of \( \mathcal{L}_2 \) to 1 often leads to good performance in our empirical studies.

Finally, we can predict the probability that user \( u \) will purchase item \( i \) at the \((\ell + 1)\)th step as follows,

\[ \hat{o}_{\ell+1,i} = z_{\ell}(h_i)^T. \]  

(17)

where \( z_{\ell} = x_{u,\ell} + g'_{u,\ell} + u_u \). Note that \( x_{u,\ell} \) and \( g'_{u,\ell} \) are optimized in \( \mathcal{L}_1 \) while \( u_u \) is optimized in \( \mathcal{L}_2 \).
Datasets (1/4)

- We conduct offline experiments (online training and inference is an important future direction) on two public and real-world datasets in e-commerce scenarios.
- Tmall\(^1\) is released at the IJCAI Competitions 2015.
- User Behavior (UB)\(^2\) is released at the IJCAI Competitions 2016.
- Both datasets contain different types of behaviors, i.e., examination, add-to-favorite, add-to-cart and purchase.

---

\(^1\)https://tianchi.aliyun.com/dataset/dataDetail?dataId=42
\(^2\)https://tianchi.aliyun.com/dataset/dataDetail?dataId=649
We preprocess the datasets as follows:

(i) for duplicated (user, item, behavior) tuples in a sequence, we only retain the first one;

(ii) we discard the cold-start items with fewer than 10 and 20 purchase interactions for UB and Tmall, respectively;

(iii) we discard the cold-start users with fewer than 5 and 10 purchase interactions for UB and Tmall, respectively;

(iv) we remove the records of adds-to-cart in Tmall because of its rarity;
(v) for each user, we take the last two purchase interactions as the validation and test data (note that the interactions between them are kept in final evaluation), and those before the penultimate purchase as the training data; and

(vi) for the preferred items in the test data of each user, we remove them from the training data since we aim to recommend new items [Wang et al., 2020].
### Datasets (4/4)

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Users</th>
<th># Items</th>
<th>Avg. Length</th>
<th>Behavior set</th>
</tr>
</thead>
<tbody>
<tr>
<td>UB</td>
<td>20,858</td>
<td>30,853</td>
<td>33.71</td>
<td>{e, f, c, p}</td>
</tr>
<tr>
<td>Tmall</td>
<td>17,209</td>
<td>16,174</td>
<td>48.60</td>
<td>{e, f, p}</td>
</tr>
</tbody>
</table>

**Table:** Statistics of the processed datasets, where **Avg. Length** denote the average length of users’ interaction sequences in the datasets, **e, f, c and p** denote examination, add-to-favorite, add-to-cart and purchase, respectively.
We evaluate the recommendation performance via recall ($\text{Rec@N}$) and normalized discounted cumulative gain ($\text{NDCG@N}$), where $N \in \{1, 5, 10\}$.

- $\text{Rec@N}$ means the proportion of cases when the preferred item is in a top-$N$ recommendation list.
- $\text{NDCG@N}$ pays attention to whether it has a relatively high-ranking position.
Baselines (1/3)

- Two single-behavior recommendation (SBR) methods:
  - BPRMF [Rendle et al., 2009]. A non-sequential model that optimizes matrix factorization using a pairwise ranking loss.
  - FISM [Kabbur et al., 2013]. A non-sequential model that represents a user with his or her interacted items.

- Two multi-behavior recommendation (MBR) methods:
  - MB-GMN [Xia et al., 2021]. A state-of-the-art GNN-based multi-behavior model that incorporates a graph meta network to learn multi-behavior patterns.
  - VAE++ [Ma et al., 2022]. A state-of-the-art VAE-based multi-behavior model that utilizes three types of signals, including the purchases, the examinations and their mixed behaviors.
Baselines (2/3)

Six single-behavior sequential recommendation (SBSR) methods:

- **FPMC** [Rendle et al., 2010]. A classic sequential model based on matrix factorization and first-order Markov chains (MCs).
- **Fossil** [He and McAuley, 2016]. A classic sequential model which combines FISM [Kabbur et al., 2013] and high-order MCs to consider more than one previous item.
- **GRU4Rec+** [Hidasi and Karatzoglou, 2018]. An RNN-based model which improves GRU4Rec [Hidasi et al., 2016] by applying a BPR-max loss and an additional sampling strategy.
- **Caser** [Tang and Wang, 2018]. A CNN-based model that utilizes some horizontal and vertical convolutional filters to capture different sequential patterns.
- **SASRec** [Kang and McAuley, 2018]. A pioneering model based on hierarchical self-attention modules.
- **FISSA** [Lin et al., 2020]. A recent model for sequential recommendation that uses SASRec to learn a user’s local preferences, an attentive version of FISM to learn his or her global preferences, and a gating module for balancing these two parts.
Baselines (3/3)

- Five multi-behavior sequential recommendation (MBSR) methods:
  - **RIB** [Zhou et al., 2018]. An RNN-based model that takes the concatenation of the item embedding and the behavior embedding as the input of a GRU layer.
  - **BINN** [Li et al., 2018]. An RNN-based model that designs a contextual long short-term memory (CLSTM) structure to model the item and behavior sequences.
  - **MGNN-SPred** [Wang et al., 2020]. A GNN-based model for MBSR that constructs a multi-relational item graph based on all kinds of behavior-specific item sub-sequences.
  - **M-SR** [Meng et al., 2020]. A GNN-based model that uses a gated GNN (GGNN) to model the item sequences and uses a GRU layer to model the sequential patterns from sequences of operations (i.e., behavior types).
  - **ASLI** [Tanjim et al., 2020]. A recent model that uses a self-attention layer to model the item sequences, and a convolutional network to leverage the behavior and category sequences to obtain the users’ intents, which are then used to query the relevant items.
Parameter Settings (1/2)

- For **RIB, BINN and ASLI**, we implement them by TensorFlow.
- For **other methods**, we use the code released by their authors:\[^345678910].

[^3]: https://cseweb.ucsd.edu/~jmcauley/
[^4]: https://github.com/hidasib/GRU4Rec
[^5]: https://github.com/graytowne/caser_pytorch
[^6]: http://csse.szu.edu.cn/staff/panwk/publications/FISSA/
[^7]: https://github.com/ciecus/MKM-SR
[^8]: https://github.com/Autumn945/MGNN-SPred
[^9]: https://github.com/akaxlh/MB-GMN
[^10]: https://csse.szu.edu.cn/staff/panwk/publications/VAEplusplus/
For fair comparison, we fix the embedding dimension \( d \) of all models as 50 and tune all the hyper-parameters on the validation data following the suggestions in the original papers.

For our NextIP, following [Kang and McAuley, 2018, Lin et al., 2020], we set the sequence length \( L \) to 50, the batch size to 128, the dropout rate to 0.5, and adopt the Adam optimizer with a learning rate of 0.001.

The length clip \( L_{clip} \) in BSA is set to 10 and the temperature parameter \( \rho \) of the contrastive loss is set to 0.07 [He et al., 2020].

The number of self-attention layers for all kinds of item sequences are searched from \( K \in \{1, 2, 3\} \) [Kang and McAuley, 2018, Lin et al., 2020].
## Main Results (1/7)

<table>
<thead>
<tr>
<th>Model</th>
<th>UB</th>
<th>Tmall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rec@1</td>
<td>Rec@5</td>
</tr>
<tr>
<td>BPRMF</td>
<td>0.086</td>
<td>0.211</td>
</tr>
<tr>
<td>FISM</td>
<td>0.095</td>
<td>0.246</td>
</tr>
<tr>
<td>MB-GMN</td>
<td>0.094</td>
<td>0.251</td>
</tr>
<tr>
<td>VAE++</td>
<td>0.139</td>
<td>0.290</td>
</tr>
<tr>
<td>FPMC</td>
<td>0.104</td>
<td>0.257</td>
</tr>
<tr>
<td>Fossil</td>
<td>0.085</td>
<td>0.219</td>
</tr>
<tr>
<td>GRU4Rec+</td>
<td>0.225</td>
<td>0.367</td>
</tr>
<tr>
<td>Caser</td>
<td>0.187</td>
<td>0.353</td>
</tr>
<tr>
<td>SASRec</td>
<td>0.226</td>
<td>0.446</td>
</tr>
<tr>
<td>FISSA</td>
<td>0.224</td>
<td>0.493</td>
</tr>
<tr>
<td>RIB</td>
<td>0.214</td>
<td>0.390</td>
</tr>
<tr>
<td>BINN</td>
<td>0.223</td>
<td>0.402</td>
</tr>
<tr>
<td>MGNN-SPred</td>
<td>0.146</td>
<td>0.291</td>
</tr>
<tr>
<td>M-SR</td>
<td>0.224</td>
<td>0.401</td>
</tr>
<tr>
<td>ASLI</td>
<td>0.230</td>
<td>0.452</td>
</tr>
<tr>
<td>NextIP</td>
<td>0.247</td>
<td>0.509</td>
</tr>
</tbody>
</table>

Table: Recommendation performance of our NextIP and four groups of baselines on UB and Tmall. Note that the best result of each column is marked in bold, and the second best result is underlined.
### Main Results (2/7)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>BPRMF</th>
<th>VAE++</th>
<th>SASRec</th>
<th>ASLI</th>
<th>NextIP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rec@5</td>
<td>0.0143</td>
<td>0.0377</td>
<td>0.0436</td>
<td>0.0423</td>
<td><strong>0.0448</strong></td>
</tr>
<tr>
<td>UB</td>
<td>NDCG@5</td>
<td>0.0086</td>
<td>0.0250</td>
<td>0.0224</td>
<td>0.0221</td>
<td><strong>0.0231</strong></td>
</tr>
<tr>
<td></td>
<td>Rec@10</td>
<td>0.0281</td>
<td>0.0564</td>
<td>0.0766</td>
<td>0.0731</td>
<td><strong>0.0790</strong></td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.0130</td>
<td>0.0310</td>
<td>0.0331</td>
<td>0.0320</td>
<td><strong>0.0340</strong></td>
</tr>
<tr>
<td>Tmall</td>
<td>Rec@5</td>
<td>0.0094</td>
<td>0.0255</td>
<td>0.0488</td>
<td>0.0514</td>
<td><strong>0.0542</strong></td>
</tr>
<tr>
<td></td>
<td>NDCG@5</td>
<td>0.0057</td>
<td>0.0175</td>
<td>0.0271</td>
<td>0.0283</td>
<td><strong>0.0314</strong></td>
</tr>
<tr>
<td></td>
<td>Rec@10</td>
<td>0.0189</td>
<td>0.0387</td>
<td>0.0821</td>
<td>0.0859</td>
<td><strong>0.0896</strong></td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.0087</td>
<td>0.0217</td>
<td>0.0379</td>
<td>0.0394</td>
<td><strong>0.0428</strong></td>
</tr>
</tbody>
</table>

**Table:** Recommendation performance of our NextIP and four representative baselines on UB and Tmall under the full-ranking setting, in which all items are considered as candidates. Note that the best result of each column is marked in bold, and the second best result is underlined.
We can have the following observations:

- For **non-sequential single-behavior recommendation methods**, FISM beats BPRMF in most cases, which indicates that modeling of user representation from all the interacted items can often lead to performance improvement.

- For **non-sequential multi-behavior recommendation methods**, both the GNN-based method MB-GMN and VAE-based method VAE++ outperform the single-behavior recommendation methods in all cases, indicating the benefits of mining multi-behavior preferences.
Main Results (4/7)

Two state-of-the-art MBR methods is worse than deep-learning-based SBSR methods and MBSR methods. It indicates that capturing the complex sequential information (e.g., global or local preferences, behavior transitions, behavior dependency, etc.) is critical in sparse scenarios of sequence data.
Main Results (5/7)

Among the methods for **sequential recommendation**, we can see that:

- These methods surpass the non-sequential methods to a large extent in all cases, showing the **importance of modeling the sequential information**.

- SASRec and FISSA consistently perform better than Caser and GRU4Rec+ on both datasets, which is consistent with the observations in previous studies [Kang and McAuley, 2018, Lin et al., 2020] and indicates the **advantage of a self-attention network for modeling the item sequences**.
Among the methods for **multi-behavior sequential recommendation**, we can observe that:

- The RNN-based methods, i.e., RIB and BINN, beat the single-behavior sequential method GRU4Rec+.
- MGNN-SPred performs poorly in our setting, which is believed because of uncompetitiveness of GNN-based methods for sequential recommendation. This is also observed by other researchers [Chang et al., 2021].
- ASLI outperforms BINN on both datasets, especially when coping with longer sequences in Tmall.
Main Results (7/7)

- Our NextIP consistently achieves the best performance on both datasets comparing with all the baselines.
- Different from all the existing MBSR methods, our NextIP adopts a dual-task learning strategy to utilize the behavior sequences, behavior-specific and behavior-agnostic item sequences in a novel and unified way.
- In addition, we address the heterogeneity challenge in task 1 and address the sparsity challenge in task 2, leading to significantly better performance.
Ablation Study (1/4)

We conduct an ablation study to understand the contribution of different components of our NextIP:

<table>
<thead>
<tr>
<th>Architecture</th>
<th>UB</th>
<th>Tmall</th>
</tr>
</thead>
<tbody>
<tr>
<td>NextIP (w/o TBCG &amp; BSA)</td>
<td>0.556</td>
<td>0.616</td>
</tr>
<tr>
<td>NextIP (w/o TBCG)</td>
<td>0.577</td>
<td>0.635</td>
</tr>
<tr>
<td>NextIP (w/o BSA)</td>
<td>0.624</td>
<td>0.678</td>
</tr>
<tr>
<td>NextIP (w/o BSA &amp; $g_{u,\ell}$ in TBCG)</td>
<td>0.557</td>
<td>0.634</td>
</tr>
<tr>
<td>NextIP (w/o BSA &amp; $x_{u,\ell}^*$ in TBCG)</td>
<td>0.570</td>
<td>0.648</td>
</tr>
<tr>
<td>NextIP</td>
<td><strong>0.632</strong></td>
<td><strong>0.681</strong></td>
</tr>
</tbody>
</table>

**Table:** Recommendation performance (Rec@10) of our NextIP with different architectures on UB and Tmall for ablation studies.
Ablation Study (2/4)

We have the following observations:

- **By removing TBCG** in task 1 and the **BSA** mechanism in task 2, our NextIP reduces to SASRec, i.e., NextIP(w/o TBCG&BSA), which only contains task 1 and utilizes the behavior-agnostic item sequences.

- The **improvement** of NextIP(w/o TBCG) over NextIP(w/o TBCG&BSA) demonstrates the **rationality of constructing the purchase prediction task** to exploit the heterogeneous transitions from auxiliary behaviors to target behavior in a user’s perspective.

- The performance gap between NextIP(w/o TBCG&BSA) and NextIP(w/o BSA) indicates that transferring the unique knowledge of different types of behaviors to help predict the next interaction in a behavior-aware manner can **improve the performance**.
Comparing NextIP(w/o BSA) with NextIP(w/o BSA & $g_{u,\ell}$ in TBCG), i.e., the variant of NextIP(w/o BSA) that removes $g_{u,\ell}$ in TBCG and only uses $x_{u,\ell}^*$, the performance decreases showing the usefulness of the target behavior embedding in balancing the context information from different behaviors in our TBCG. Similarly, we can see the necessity of using $x_{u,\ell}^*$ to refine the final context representation by comparing NextIP(w/o BSA) with NextIP(w/o BSA & $x_{u,\ell}^*$ in TBCG).
Ablation Study (4/4)

- NextIP(w/o BSA & $g_{u,\ell}$ in TBCG) perform worse than NextIP(w/o BSA & $x_{u,\ell}^*$ in TBCG), the reason might be that when using only single context information in predicting different behaviors, the behavior-specific item sequence encoders are hard to get trained well.

- Our NextIP performs the best compared with all of its variants, which clearly demonstrates the positive complementary effect of all the designed components in our NextIP.
We conduct an additional ablation study by removing each kind of behavior-specific item sequences in task 1, we can get four variants of our NextIP, i.e., “-e”, “-f”, “-c”, “-p”.
Impact of Behavior-Specific Item Sequences (2/3)

Figure: Recommendation performance (Rec@10) of our NextIP and its variants by removing different behavior-specific item sub-sequences on Tmall and UB.

Luo et al., (Shenzhen University)
Impact of Behavior-Specific Item Sequences (3/3)

We have the following observations:

- The performance decline of removing the examination-specific item sequences (i.e., “-e”) is more significant than that of “-f” or “-c” on both datasets, because the examinations are much more than others.

- Our NextIP performs the best compared with all of its variants, which again clearly demonstrates the effectiveness and rationality of utilizing all the behavior-specific item sequences in our NextIP.
Case Study (1/3)

- We obtain the **attention scores** of all the users when predicting different target behaviors in the training set.
- We visualize the **mean attention scores** from all the users in each dataset as a heatmap matrix.
- Note that the horizontal axis represents the **source behaviors** and the vertical axis represents the **target behaviors**.
Case Study (2/3)

Figure: Visualization of the attention scores of different source behaviors to different target behaviors in target-behavior aware context aggregator (TBCG) of our NextIP on UB and Tmall.

(a) UB

(b) Tmall
We have the following observations:

- The **main diagonal elements** of both two heatmap matrices have the **lowest attention scores** in each corresponding row, which is reasonable since $\mathbf{x}^{*,\ell}_{u,\ell}$ is added to the final context representation as shown in Eq.(9).

- When **predicting purchases**, the information contribution of adds-to-favorite and adds-to-cart is larger than that of examinations on both datasets, which is consistent with our intuition in online shopping.

- When **predicting adds-to-cart**, the behaviors of adds-to-favorite are more informative than purchases in providing knowledge, because the interests reflected in the purchase-specific item sequences may have **expired** to some extent. In general, these observations further demonstrate the **rationality and interpretability** of our NextIP.
Conclusions

- We propose a novel framework named NextIP for multi-behavior sequential recommendation, which adopts a dual-task learning strategy to convert the problem to a next-item prediction task and a purchase prediction task.

- We design a target-behavior aware context aggregator (TBCG) to transfer the unique knowledge of different behaviors so as to predict the next interaction in a behavior-aware manner more accurately.

- We design a behavior-aware self-attention (BSA) mechanism to aggregate the collection of historical interacted items of each behavior and treat them as negative samples for more accurate learning of purchase-oriented preference.
For future works, we are interested in extending our NextIP to incorporate knowledge graphs about users and items.
Thank you!

- We thank the support of National Natural Science Foundation of China Nos. 62172283 and 61836005. We thank Miss Jing Lin for relevant data preprocessing scripts and Mr. Dugang Liu for his helpful discussions.
- If you have any questions, please feel free to contact us.
Sequential recommendation with graph neural networks.

Momentum contrast for unsupervised visual representation learning.

Fusing similarity models with Markov chains for sparse sequential recommendation.

Recurrent neural networks with top-k gains for session-based recommendations.

Session-based recommendations with recurrent neural networks.
In Proceedings of the 4th International Conference on Learning Representations.

FISM: Factored item similarity models for top-N recommender systems.
In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 659–667.

Self-attentive sequential recommendation.

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.

FISSA: Fusing item similarity models with self-attention networks for sequential recommendation.

VAE++: Variational autoencoder for heterogeneous one-class collaborative filtering.
In Proceedings of the 15th International Conference on Web Search and Data Mining, pages 666–674. ACM.

Meng, W., Yang, D., and Xiao, Y. (2020).
Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation.

BPR: Bayesian personalized ranking from implicit feedback.

Factorizing personalized Markov chains for next-basket recommendation.

Personalized top-N sequential recommendation via convolutional sequence embedding.

Attentive sequential models of latent intent for next item recommendation.

Attention is all you need.

Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction.

Graph meta network for multi-behavior recommendation.

Deep feedback network for recommendation.

Graph neural network based model for multi-behavior session-based recommendation.
volume 26, pages 429–447.

Micro behaviors: A new perspective in e-commerce recommender systems.