A Generic Behavior-Aware Data Augmentation Framework for Sequential Recommendation

Jing Xiao
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China
2200271039@email.szu.edu.cn

Weike Pan*
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China
panweike@szu.edu.cn

Zhong Ming
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China
mingz@szu.edu.cn

ABSTRACT

Multi-behavior sequential recommendation (MBSR), which models multi-behavior sequentiality and heterogeneity to better learn users' multifaceted intentions has achieved remarkable success. Though effective, the performance of these approaches may be limited due to the sparsity inherent in a real-world data. Existing data augmentation methods in recommender systems focus solely on a single type of behavior, overlooking the variations in expressing user preferences via different types of behaviors. During the augmentation of samples, it is easy to introduce excessive disturbance or noise, which may mislead the next-item recommendation. To address this limitation, we propose a novel generic framework called multi-behavior data augmentation for sequential recommendation (MBASR). Specifically, we design three behavior-aware data augmentation operations to construct rich training samples. Each augmentation operation takes into account the correlations between behaviors and aligns with the users' behavior patterns. In addition, we introduce a position-based sampling strategy that can effectively reduce the perturbation brought by the augmentation operations to the original data. Note that our model is data-oriented and can thus be embedded in different downstream MBSR models, so the overall framework is generic. Extensive experiments on three real-world datasets demonstrate the effectiveness of our MBASR and its applicability to a wide variety of mainstream MBSR models. Our source code is available at https://github.com/XiaoJing-C/MBASR.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Sequential Recommendation; Data Augmentation; Multi-Behavior Modeling

*Corresponding author

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07...$15.00
https://doi.org/10.1145/3626772.3657682

ACM Reference Format:
Jing Xiao, Weike Pan, and Zhong Ming. 2024. A Generic Behavior-Aware Data Augmentation Framework for Sequential Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3626772.3657682

1 INTRODUCTION

Recommender systems have become an indispensable part of various online platforms (e.g., e-commerce, streaming services, and social media), which can greatly alleviate the problem of information overload and help users discover items that meet their interests [9, 12, 19]. Since the user preferences shift dynamically as time goes on, the effective utilization of the order of the user-item interactions has become increasingly crucial. Sequential recommendation (SR) aims to predict the next item that a user is most likely to interact with based on his or her historical interaction sequence, which has emerged as a research hotspot in recent years [1, 6, 25, 37]. However, most existing SR models generally focus on a single type of behavior, overlooking the valuable information among heterogeneous behaviors. This limitation results in insufficient learning of user preferences, which sparks research into the problem of multi-behavior sequential recommendation (MBSR).

The mainstream research on MBSR aims to enhance the understanding of sequential patterns in diverse user behaviors. With the prosperity of deep neural networks, recurrent neural network (RNN)-based [3, 16, 45], graph neural network (GNN)-based [2, 15, 23, 34], and Transformer-based models [7, 28, 31, 42] have emerged as the major branches. Despite yielding promising results, these methods still suffer from the issue of sparsity inherent in a real-world data. Due to the lack of comprehensive user behavior information, models often struggle to accurately capture complex collaborative signals, thus hindering model training. To tackle this problem, data augmentation methods have been proposed and attracted many researchers.

In recent years, inspired by the success of self-supervised learning (SSL), many studies have begun to use contrastive learning (CL) to enhance training samples. S^3-Rec [44] incorporates the item attribute information as an additional data and captures the correlations between item-attribute, sequence-item, sequence-attribute, and sequence-subsequence using maximum mutual information (MMI). CL4SRec [39] introduces three data augmentation operations to construct different augmented views and applies contrastive learning to minimize the differences between different augmented views of a same sequence. DuoRec [26] proposes a model-level data augmentation method based on different dropout masks, along with an additional supervised approach to construct positive samples. Another promising research line is to expand the original sequences through some pre-trained methods. For instance, CASR [35] first pre-trains a sampling model and an anchor model, and then constructs some counterfactual sequences to re-optimize the anchor model via the sampling model. Both ASRep [21] and BiCAT [13] rely on Transformer to reversely generate new items to address the issue of short sequences. In addition, RSS [24] suggests that each item in a sequence should be possible to be sampled as a target item, and designs a recency-based sampling strategy to generate some different new samples.

Despite the effectiveness of these data augmentation methods, there are still some limitations: (1) These methods only focus on a single type of behavior, neglecting the correlations between multi-type behaviors in user-item interactions, which has twofold drawbacks. Firstly, models that extend the original sequences through pre-training (e.g., CASR and ASRep) tend to generate items deviating from user preferences, as they fail to adequately exploit the information between different types of behaviors. Secondly, failure to differentiate behavior types when constructing new samples may introduce excessive noise. Empirically, auxiliary behaviors and target behaviors exhibit varying degrees of influence in expressing user preferences. However, models like CL4SRec, when generating positive samples through random transformations of purchased items, may inadvertently introduce significant noise to the original sequences, thereby impairing the model's performance. (2) Enhancing training samples with some pre-trained models often relies on a specific network architecture, such as Transformer or GNN (e.g., BiCAT and GraphDA), which makes it challenging to seamlessly transfer to other network structures. Besides, it requires extra computational resources, which may pose constraints in practical application scenarios. (3) Models like S^3-Rec rely on some additional auxiliary information while limited data fail to provide sufficient support. To overcome these limitations, we propose a novel multi-behavior data augmentation model for sequential recommendation, namely MBASR for short.

In contrast to most data augmentation approaches, our work exclusively focuses on the data level, without involving any changes to the original model structure or loss function, representing a completely **non-intrusive** and **data-oriented** model. Therefore, it can be generalized to a wide range of MBSR models. It is well understood that before a user decides to purchase an item, he or she will click many related items. And when the purchase behavior is completed, it often indicates that his or her current preference has been satisfied. Thus, the short sequence from clicking to purchasing can be regarded as the user's short-term preference in a specific period. We use the purchase behavior as a marker to divide a sequence into some subsequences to get a more fine-grained sample. With the subsequences, we design specific behavior-aware data augmentation operations within and between subsequences to generate new samples, including order perturbation (OP), redundancy reduction (RR), and pairwise swapping (PS). Notably, order perturbation and redundancy reduction operate on the item level, while pairwise swapping operates on the subsequence level. Furthermore, inspired by RSS[24], we devise a position-based sampling strategy, where items closer to the current item have a lower probability of being sampled. This mitigates the risk of introducing excessive noise in new samples and thus jeopardizes model performance. In the real-world scenario, user-item interactions are often dynamic and diverse. The data augmentation operations we designed to broaden the sequential patterns to simulate the inherent randomness and variability in user behaviors. Consequently, the model can acquire more generalized features, effectively adapting to previously unseen data and enhancing robustness.

We summarize our main contributions as follows:

- To the best of our knowledge, this is the first data augmentation method specifically tailored to solve the sparsity problem in MBSR.
- We propose three behavior-aware data augmentation methods, including order perturbation, redundancy reduction, and pairwise swapping, to generate new sequences. These methods can be seamlessly integrated into various existing MBSR models to boost their performance.
- We design a position-based sampling strategy that can maximally preserve the smooth sequentiality between neighboring items in the original sequences.
- We conduct extensive experiments on three real-world datasets to demonstrate the significant superiority of our MBASR over baseline models and its applicability to a wide variety of mainstream MBSR models.

2 RELATED WORK

In this section, we review some mainstream models for sequential recommendation and some representative data augmentation methods for recommendation.

2.1 Sequential Recommendation

2.1.1 Single-behavior Sequential Recommendation. Early works on single-behavior sequential recommendation (SBSR) mostly rely on the Markov chains (MCs) to capture the dependencies of items in a sequence. For example, FPMC [30] combines the power of matrix factorization (MF) and first-order MCs to model both general tastes and sequential patterns. Based on this idea, some improved versions with high-order MCs have been proposed to consider more previous actions. Fossil [8] integrates higher-order MCs and factored item similarity to capture long-term and short-term dynamics.

With the boosting of deep learning, RNN has made great achievements in processing sequential data, and a series of models based on RNN and its variants have emerged in sequential recommendation [4, 10, 11, 29]. Among them, GRU4Rec [10] is a pioneering work that applies RNN to session-based recommendation. It adopts GRU to obtain a user's representations and utilizes session-parallel mini-batches to speed up training. In addition to RNN, some other deep learning models also have been designed for sequential recommendation. Convolutional neural network (CNN)-based method Caser [32] learns rich sequential patterns by sliding horizontal and vertical convolution filters to get a user's interests. SASRec [14] leverages the self-attention mechanism to gain item-to-item correlations and achieves the state-of-the-art performance.

FISSA [17] extracts a user's long-term and short-term preferences by combining the self-attention mechanism and item similarity model. Furthermore, GNN has been designed to model complex transitions between adjacent items by constructing sequences of user-item interactions into graph structures, and then learning the representation of each node to mine the informative context information [22, 27, 38, 40]. For example, SR-GNN [38] proposes to apply a gated graph neural network (GGNN) and a self-attention network to model sessions. However, these models only consider a single type of user behavior and do not sufficiently extract knowledge from the user-item interaction sequences, thus largely missing some valuable information.

2.1.2 Multi-behavior Sequential Recommendation. Researches on MBSR aim to exploit the dependencies across different types of user behaviors to better infer users' dynamic preferences for target behavior recommendation. There are relatively few studies on MBSR, which are mostly based on deep learning. RLBL [18] divides a heterogeneous sequence into some time windows, combines the advantages of RNN and LBL to capture a user's long-term and short-term preferences, respectively, and introduces some behavior-specific transition matrices to model different types of behaviors. RIB [45] encodes each item and behavior in a user's historical interaction sequence into a corresponding embedding vector, captures the sequential information through GRU, and uses an attention layer to obtain the varied effects of different types of behaviors. BINN [16] proposes a novel context-aware LSTM network, which can simultaneously memorize the item and behavior information in a heterogeneous sequence. BAR [7] designs a behavior attention layer and a task-specific layer to learn the relationship between target behavior and items with different types of behaviors, which can adapt an SBSR model to an MBSR model. DyMus [3] decomposes an entire sequence into some independent behavior subsequences and learns sequence representations through dynamic routing. However, for a sparse data, partitioning a sequence into some behavior-specific subsequences leads to insufficient information in each subsequence. This hinders dynamic routing from providing an adequate contextual understanding for precise user interest prediction, leading to suboptimal performance. Hence, we do not include it in the experimental section.

Furthermore, another promising research line is adopting GNN to capture complex behavior transition patterns between items [2, 23, 34, 41]. For example, MKM-SR [23] uses a gated GNN (GGNN) and a GRU layer to learn item features and behavior representations at each time step, respectively. MGNN-SPred [34] constructs a multi-relational item graph (MRIG) to capture the purchase-to-purchase and click-to-click relationships and learn a user's intention. GPG4HSR [2] captures the transition between behaviors through a global graph (GG) and introduces a user-specific personalized graph (PG) to mine a user's personalized interests. Moreover, it uses a global personalized fusion layer (GPF) to integrate the global and personalized information.

2.2 Data Augmentation for Recommendation

The aforementioned well-known studies primarily focus on designing various network structures to reveal user preferences. However, these methods may achieve suboptimal performance in on a sparse data, for which, data augmentation becomes particularly crucial.

With the remarkable success of contrastive learning in recommender systems, a bunch of models leveraging contrastive learning for data augmentation have emerged. CL4SRec [39] proposes three augmentation operations, i.e., crop, mask and reorder. It treats the augmented sequences constructed from the same sequence as positive pairs and employs a Transformer-based model to encode user representations. CoSeRec [20] builds upon CL4SRec by introducing two informative augmentation operations, which involves replacing and inserting items with related items. DuoRec [26] constructs positive and negative samples through both unsupervised and supervised approaches. It utilizes different dropout masks to generate semantically similar but feature-distinct positive samples, and assumes that sequences with the same target item contain similar user preferences. MoCo4SRec [36] integrates the momentum updating mechanism to sequential recommendation for the first time. However, these models are based on the Transform architecture, which is not well suited for short sequences. Another latest line of research relies on some pre-trained models to extend or enhance sequences. L2Aug [33] categorizes users into casual users and core users, and assists the casual users through the core users. ASRep [21] and BiCAT [13] fill a sequence by reversely generating prior items, which does not change the original sequence and preserves the information of the original data to the maximum extent. GraphDA [5] first pre-trains a graph-based model to obtain useful user and item representations, and then employs a top-k sampling strategy to enhance the user-item matrix. Moreover, RSS [24] considers that every item in a sequence should be possible as the target item. It designs a recency-based sequence sampling strategy to reconstruct the training samples, representing a completely model-agnostic approach.

In the existing studies, we note that there is no augmentation method specifically designed for MBSR, and still faces challenges from short sequences. Inspired by the above observations, we propose a generic behavior-aware data augmentation framework tailored for MBSR. Similar to RSS [24], our work concentrates solely on the data level and is a model-agnostic and non-intrusive approach. Taking a multi-behavior sequence perspective, we design three behavior-aware data augmentation operations. These operations aim to preserve the information from the original sequences as much as possible while generating new training samples that align with the user behavior patterns.

3 PRELIMINARIES

In this section, we first formulate the studied problem and then give an overview of our proposed framework, i.e., multi-behavior data augmentation sequential recommendation (MBASR).

3.1 Problem Definition

Let $\mathcal{U} = \{u_1, u_2, \cdots, u_{|\mathcal{U}|}\}$ and $\mathcal{V} = \{v_1, v_2, \cdots, v_{|\mathcal{V}|}\}$ represent the sets of users and items, respectively, where $|\mathcal{U}|$ and $|\mathcal{V}|$ denote the numbers of users and items. We consider there are $|\mathcal{B}|$ types of behaviors denoted by $\mathcal{B} = \{b_1, b_2, \cdots, b_{|\mathcal{B}|}\}$. Each user $u \in \mathcal{U}$ is associated with an interaction sequence of (item, behavior) pairs in chronological order $\mathcal{S}_u = \{(v_u^1, b_u^1), \cdots, (v_u^\ell, b_u^\ell), \cdots, (v_u^L, b_u^L)\}$,

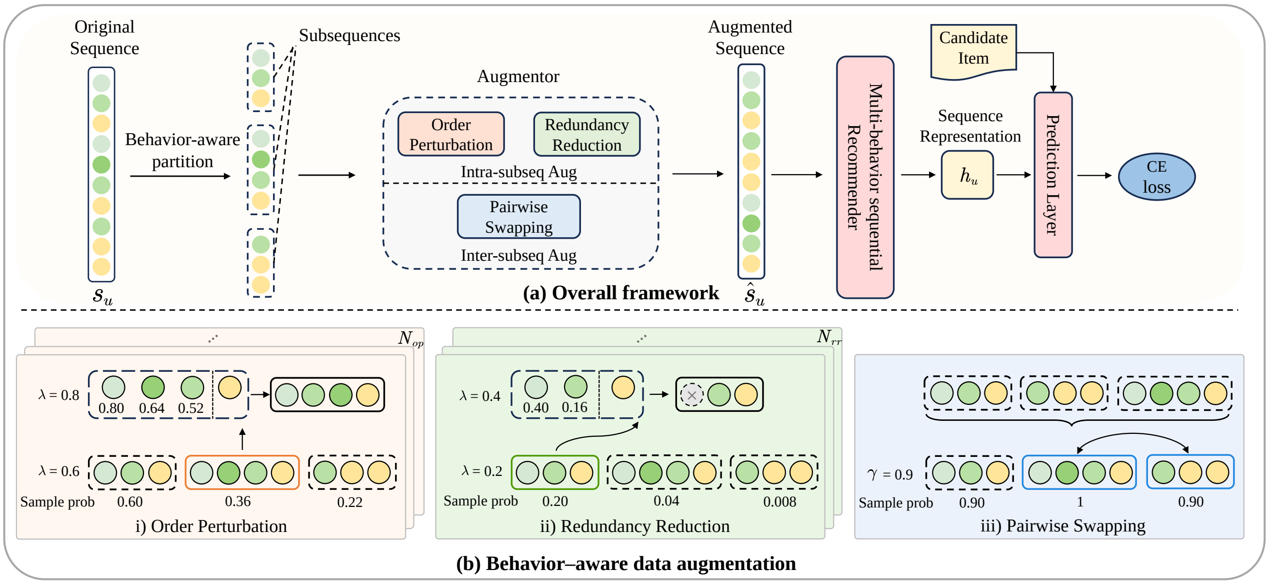

Figure 1: The overall architecture of the proposed MBASR model.

where $v_u^\ell \in \mathcal{V}$ denotes the ℓ-th item interacted by user u with behavior $b_u^\ell \in \mathcal{B}$.

Given the multi-behavior interaction sequences of all users $\mathcal{S}$, the set of users $\mathcal{U}$, the set of items $\mathcal{V}$ and the set of behavior types $\mathcal{B}$, MBSR seeks to predict the most likely-to-purchase item of a user u from $\mathcal{V}$ at time step $L+1$, i.e., v_u^{L+1}, which can be formulated as follows:

$$\arg\max_{v_u^i \in \mathcal{V}} P\left(v_u^{L+1} = v_u^i \mid \mathcal{S}_u\right) \tag{1}$$

3.2 Overview

To address the data sparsity problem that exists in multi-behavior sequences, we propose a framework that is generic to different MBSR models to boost their performance and named it MBASR. The basic idea of our MBASR is to design different behavior-aware data augmentation operations to generate rich training samples that provide insights into users' dynamic preferences, which can be applied to various downstream tasks. Our framework takes the user-item interaction sequence $\mathcal{S}_u$ as input and aims to generate an augmented sequence $\hat{\mathcal{S}}_u$ through an augmentor. Specifically, we take the purchase behavior as the boundary to divide the original sequence into multiple subsequences. Then, we consider the impact of different types of behaviors on modeling user preferences, applying a series of transformations to the items within and between these subsequences. In addition, we use some specific sampling strategies to minimize the noise from random transformations. Finally, we use $\hat{\mathcal{S}}_u$ as input to a downstream MBSR task to achieve more precise recommendation results. The overview of our MBASR is illustrated in Figure 1, and more details will be introduced in Section 4.

4 METHODOLOGY

In this section, we will introduce each component of our MBASR in detail.

4.1 Behavior-aware Partition

In practical application scenarios, users usually click, favorite and add to cart multiple related items before making a purchase. Therefore, each purchase action and the auxiliary behaviors between two consecutive purchases can be regarded as a user's short-term preference within a specific period. Based on this observation, we divide an original sequence of user-item interactions $\mathcal{S}_u$ into some subsequences to gain a better comprehension of the user's purchase behavior and short-term interest tendencies. It should be noted that in our study, we currently only consider one auxiliary behavior, clicking, due to its accessibility and relatively higher frequency.

In Figure 1 (a), we depict the process in which subsequences are partitioned, where yellow circles symbolize purchase and green circles denote click. It's evident that the complete user-item interaction sequence has been segmented into three subsequences, with each one starting with a click and ending with a purchase.

Here, we omit the superscript u for each item v and the behavior representation b for simplicity, and $\mathcal{S}_u$ can thus be further represented as follows:

$$\mathcal{S}_u = \{v^1, \cdots, v^\ell, \cdots, v^L\} = \{s^1, s^2, \cdots, s^m\} \tag{2}$$

where s denotes the subsequence composed of multiple items, and m denotes the total number of subsequences within $\mathcal{S}_u$.

4.2 Position-based Sampling

We hypothesize that items closer to the current position inherently align more with the present user's preferences and, therefore, exert a more substantial influence on a downstream task. As a result, our sampling strategy hinges on the order of item positions, with items closer to the current position assigning a lower probability of being sampled. We denote $g(\cdot)$ as the score function, which assigns higher scores to items that hold lower positional significance. At the k-th position, $g(k)$ can be formulated as follows:

$$g(k) = \lambda^k \tag{3}$$

And we can get the sampling probability $f(k)$ by:

$$f(k) = g(k) / \sum_{\ell=0}^{L-1} g(\ell) \tag{4}$$

Here, $0 \leq \lambda \leq 1$ serves as the weighting coefficient that modulates the importance of different positions, thereby influencing the magnitude of the sampling probabilities. When $\lambda \longrightarrow 0$, items at the beginning of a sequence are more likely to be sampled. When $\lambda \longrightarrow 1$, the probability of each item being sampled in the sequence tends to be uniform.

4.3 Behavior-aware Data Augmentation

In this section, we present our three proposed data augmentation methods for MBSR, i.e., subseq order perturbation (OP), subseq redundancy reduction (RR), and pairwise subseq swapping (PS). Among them, both subseq order perturbation and subseq redundancy reduction occur within subsequences, summarized as Intra-subseq Augmentation, while pairwise subseq swapping takes place between subsequences, referred to as Inter-subseq Augmentation. We illustrate the three data augmentation operations in Figure 1 (b), and finally summarize the complete learning algorithm of our framework in Algorithm 1.

4.3.1 *Intra-subseq Augmentation.*

- **Order Perturbation (OP)**. In the study of user shopping habits, it is observed that users often click a series of similar items before making a purchase. Due to the existence of item similarity, these click behaviors do not strictly adhere to a specific order, and the correlations between successive clicked items are not particularly strong. Based on this observation, we propose to inject controlled order perturbations into the clicked items within subsequences to introduce diverse sequential patterns.
Specifically, we first calculate the sampling probability f_s for each subsequence in the sequence $\mathcal{S}_u$ based on Equations (3) and (4). Then, we sample a certain subsequence $\underline{s}^{op}$ from $\mathcal{S}_u$ using the computed probabilities f_s. To keep the operation simple, within $\underline{s}^{op}$, we again employ Equations (3) and (4) to determine the sampling probability f_v for each clicked item. Finally, an order perturbation operation is performed based on f_v, which means that each clicked item in $\underline{s}^{op}$ is reordered according to the value of f_v. Notably, since the calculation of f_s and f_v follows the idea mentioned above that items closer to the current item have a smaller probability of being sampled, the perturbation applied to the items in $\underline{s}^{op}$ is designed to maintain their original relative positions as much as possible, rather than being randomly shuffled. The new sequence can be formulated as:

$$\mathcal{S}_u^{OP} = OP(\mathcal{S}_u) = \left\{s^1, s^2, \cdots, \underline{s}^{op}, \cdots, s^m\right\} \tag{5}$$

$$\underline{s}^{op} = \{\hat{v}_1, \hat{v}_2, \cdots, \hat{v}_c, \cdots v_n\} \tag{6}$$

where m and n denote the number of subsequences in $\mathcal{S}_u$ and the length of $\underline{s}^{op}$, respectively. Additionally, c denotes the number of clicked items in $\underline{s}^{op}$.
Moreover, we iterate the above operation $N_{op} = \lfloor \alpha * m \rfloor$ times, i.e., we have collectively perturbed a percentage α of subsequences, where $0 \leq \alpha \leq 1$ and $\lfloor \cdot \rfloor$ is the flooring function. Ultimately, we can obtain the augmented complete sequence $\hat{\mathcal{S}}_u^{OP}$.

- **Redundancy Reduction (RR)**. When users click items in a short period, similarities or redundancies will inevitably occur, e.g., users may explore several similar T-shirts before making a final purchase decision. Given this phenomenon, we believe that user representations can be enhanced by mitigating redundancy. On the one hand, this can preserve the user's preference as much as possible. On the other hand, intentionally reducing redundancy allows the model to capture skip-level sequential patterns, which can strengthen the model's learning capability. In pursuit of this, we choose the simple operation of deleting the specific clicked items. By randomly removing some similar or redundant items, we can simulate the randomness and diversity inherent in a user's decision-making process. Furthermore, this approach helps to prevent the model from overly relying on the historical user behavior data, and such variation will encourage the model to better adapt to potential changes in user preferences and the emergence of new interests.
Specifically, we first sample a subsequence $\underline{s}^{RR}$ based on Equations (3) and (4), and then to further mitigate the information loss due to the deletion of items, we again use Equations (3) and (4) to sample an item to be deleted. It is empirically known that purchased items contain rich valuable information, so we should minimally vary the purchased items. Here we only delete the clicked items before the purchase, and the new sequence can be formulated as:

$$\mathcal{S}_u^{RR} = RR(\mathcal{S}_u) = \left\{s^1, s^2, \cdots, \underline{s}^{rr}, \cdots, s^m\right\} \tag{7}$$

$$\underline{s}^{rr} = \{\hat{v}_1, \hat{v}_2, \cdots, \hat{v}_{c-1}, \cdots, v_{n-1}\} \tag{8}$$

Similar to the order perturbation operation, we use $N_{rr} = \lfloor \beta * m \rfloor$ to control the number of times the above operation is repeated to obtain the augmented sequence $\hat{\mathcal{S}}_u^{RR}$, where $0 \leq \beta \leq 1$.

4.3.2 *Inter-subseq Augmentation.*

- **Pairwise Swapping (PS)**. Based on the assumption above that a subsequence characterizes a user's interests during a specific period, we take it as the smallest unit, allowing for a moderate shuffling of the order between subsequences. In addition, to achieve a trade-off between the diversity of data samples and model performance, we choose to perform a swap operation on only one pair of subsequences.
Firstly, we randomly select a *source* subsequence $\underline{s}^{idx_{source}}$ from sequence $\mathcal{S}_u$ and obtain its index idx_{source}. To minimize noise amplification during the swap operation process, we design a sampling probability $p(\cdot)$ based on the position awareness of the *source* subsequence. Specifically, subsequences closer to the *source* subsequence have a higher probability of being sampled as the *destination* subsequence $\underline{s}^{idx_{dest}}$. For the k-th subsequence, its sampling probability, denoted as $p(k)$, is formalized as:

$$p(k) = \frac{x(k)}{\sum_{\ell=0}^{L-1} x(\ell)} \tag{9}$$

$$x(k) = \gamma^{|k - idx_{source}|} \tag{10}$$

Here, $0 \leq \gamma \leq 1$ is a parameter that controls the probability distribution. This relative position sampling strategy ensures a higher priority for positions closer to the *source* subsequence when sampling the *destination* subsequence. Given the original sequence $\mathcal{S}_u$, once the *source* subsequence and the *destination* subsequence are identified, we enhance $\mathcal{S}_u$ by swapping these two subsequences, resulting in the augmented sequence $\hat{\mathcal{S}}_u^{PS}$.

$$\mathcal{S}_u = \left\{s^1, \cdots, \underline{s}^{idx_{source}}, \cdots, \underline{s}^{idx_{dest}}, \cdots, s^m\right\} \tag{11}$$

$$\hat{\mathcal{S}}_u^{PS} = PS(\mathcal{S}_u) = \left\{s^1, \cdots, \underline{s}^{idx_{dest}}, \cdots, \underline{s}^{idx_{source}}, \cdots, s^m\right\} \tag{12}$$

Algorithm 1: MBASR Algorithm

input : dataset Υ, MBSR model $\mathcal{R}$, hyper-parameters α, β, γ and λ, training epochs M and batch size B.
output: $\mathcal{R}(\cdot)$.

for $epoch = 1$ *to* M **do**
  sample a minibatch $\{\mathcal{S}_u\}_{u=1}^{B}$ from Υ;
  for $u \in \{1, 2, \cdots, B\}$ **do**
    `// behavior-aware partition (Section 4.1)`
    m ← the number of subsequences in $\mathcal{S}_u$;
    Order Perturbation:
    $N_{op} = math.floor(\alpha * m)$;
    for $i = 1$ *to* N_{op} **do**
      Derive $\underline{s}^{op}$ by Equations (3-4);
      Reorder all clicked items in $\underline{s}^{op}$ by Equations (3-4);
    $\hat{\mathcal{S}}_u^{OP} \leftarrow N_{op}$ times reorder completed;
    Redundancy Reduction:
    $N_{rr} = math.floor(\beta * m)$;
    for $i = 1$ *to* N_{rr} **do**
      Derive $\underline{s}^{rr}$ by Equations (3-4);
      Delete a clicked item in $\underline{s}^{rr}$ by Equations (3-4);
    $\hat{\mathcal{S}}_u^{RR} \leftarrow N_{rr}$ times delete completed;
    Pairwise Swapping:
    Randomly select a subsequence $\underline{s}^{idx_{source}}$;
    Derive idx_{dest} by bringing idx_{source} into Equations (9-10);
    $\hat{\mathcal{S}}_u^{PS} \leftarrow$ Swap $\underline{s}^{idx_{source}}$ and $\underline{s}^{idx_{dest}}$ in $\mathcal{S}_u$;
  Train $\mathcal{R}$ based on $\hat{\mathcal{S}}_u$.

4.4 Sequential Recommender

With data augmentation operations, we construct diverse training samples for the model to better learn the evolving preferences of users. After acquiring the augmented sequence $\hat{\mathcal{S}}_u$, we feed it into an MBSR model to derive the user representation $\boldsymbol{h}_u$ as follows:

$$\boldsymbol{h}_u = \mathcal{H}(\hat{\mathcal{S}}_u) \tag{13}$$

where $\mathcal{H}(\cdot)$ is an MBSR model, aiming to effectively capture complex behavior dependencies by modeling heterogeneous behaviors to better portray dynamic user interests.

Table 1: Statistics of the processed datasets.

Dataset	#Users	#Items	#Clicks	#Purchases	Avg. length
Tmall	17,209	16,177	446,442	223,265	8.89
UB	20,858	30,793	470,731	136,250	5.82
JD	11,367	12,266	131,298	75,774	16.16

4.5 Prediction and Model Optimization

After obtaining the sequence representations and the item embedding, for each candidate item $v_i \in \mathcal{V}$, we calculate a user's preference score to it by:

$$z_i = \boldsymbol{h}_u \boldsymbol{e}_{v_i}^T \tag{14}$$

where $\boldsymbol{e}_{v_i}$ is the embedding of item v_i.

Then we apply a softmax function to normalize the preference score to get the predicted probability that user u will purchase item v_i at next time as:

$$\hat{y}_i = \frac{\exp(z_i)}{\sum_{v_j \in \mathcal{V}} \exp(z_j)} \tag{15}$$

Finally, to perform the model optimization, we minimize the cross-entropy loss function $\mathcal{L}$ between the prediction and ground truth, which can be defined as follows:

$$\mathcal{L} = -\sum_{i=1}^{|\mathcal{V}|} \delta(v_i) \log(\hat{y}_i) \tag{16}$$

where the indicator function $\delta(v_i) = 1$ only if item v_i is the true interacted item of user u at the next time step, and $\delta(v_i) = 0$ otherwise.

5 EXPERIMENTS

In this section, we first present our experimental setup. Then, we conduct extensive experiments to answer the following research questions (RQs).

- **RQ1:** How does integrating our MBASR into different mainstream multi-behavior sequential recommendation models perform compared with the original ones and the state-of-the-art data augmentation model?
- **RQ2:** How do the different data augmentation operations proposed in our MBASR enhance the performance of downstream models?
- **RQ3:** How will the data sparsity affect the performance of our MBASR?
- **RQ4:** How is the training efficiency of our MBASR compared with that of the corresponding original model?

5.1 Experimental Settings

5.1.1 Datasets. We conduct our experiments on three public recommendation datasets: (1) **Tmall**[1]**:** A public e-commerce dataset released at the IJCAI Competitions 2015, which contains users' shopping logs in six months. (2) **UB**[2]**:** Another e-commerce dataset with multiple behaviors released at the IJCAI Competitions 2016. (3) **JD**[3]**:** A dataset released by the competition of JD in 2019. For these

[1]https://tianchi.aliyun.com/dataset/dataDetail?dataId = 42
[2]https://tianchi.aliyun.com/dataset/dataDetail?dataId = 649
[3]https://jdata.jd.com/html/detail.html?id=8

Table 2: Experimental results of different MBSR models with (w/) or without (w/o) our MBASR on the three datasets. The best results are boldfaced, and "Imprv." indicates the improvement on HR@10.

Datasets	Metric	RLBL		RIB		BINN		GRUBAR		SASBAR		GPG4HSR	
		w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/
Tmall	HR@10	0.0238	**0.0300**	0.0341	**0.0349**	0.0289	**0.0299**	0.0322	**0.0359**	0.0493	**0.0539**	0.0494	**0.0531**
	NDCG@10	0.0121	**0.0148**	0.0183	**0.0185**	0.0152	**0.1560**	0.0170	**0.0188**	0.0283	**0.0296**	0.0263	**0.0280**
	Imprv.		**26.05%**		**2.35%**		**3.46%**		**10.31%**		**9.33%**		**7.49%**
UB	HR@10	0.0396	**0.0424**	0.0310	**0.0315**	0.0339	**0.0372**	0.0345	**0.0382**	0.0420	**0.0494**	0.0487	**0.0503**
	NDCG@10	0.0177	**0.0192**	0.0157	**0.0158**	0.0178	**0.0196**	0.0183	**0.0201**	0.0220	**0.0261**	0.0257	**0.0277**
	Imprv.		**7.07%**		**1.61%**		**9.73%**		**9.69%**		**17.62%**		**3.29%**
JD	HR@10	0.1387	**0.1975**	0.3196	**0.3215**	0.3174	**0.3279**	0.3048	**0.3135**	0.3085	**0.3198**	0.3235	**0.3292**
	NDCG@10	0.0621	**0.0939**	0.1797	**0.1797**	0.1825	**0.1866**	0.1743	**0.1787**	0.1652	**0.1718**	0.1789	**0.1871**
	Imprv.		**42.39%**		**0.59%**		**3.31%**		**2.85%**		**3.66%**		**1.76%**

datasets, we only keep the clicks and purchases [43], and preprocess them as follows: i) we discard the cold-start items with fewer than 10 records in UB, and 20 in Tmall and JD; ii) we discard the cold-start users with fewer than 10 records in Tmall and 5 in the other two datasets; iii) we order each dataset by the timestamps, and only keep the first (user, item, behavior) triple for repeated ones in a sequence; iv) following the LOO (leave-one-out)-based validation, in each dataset, we take the last purchase as the test set, the penultimate purchase as the validation set, and the rest as the training set; and v) to simulate the sparsity of the data, we take a portion of items at equal intervals in each user's interaction sequence for training. We process Tmall and UB at intervals of 7 and 5, respectively, and JD is not sparsified. In Section 5.4, we will discuss the impact of data sparsity on model performance. The statistics of the processed datasets are shown in Table 1.

5.1.2 Evaluation Metrics. To evaluate the recommendation performance of our proposed model, we adopt two common top-k metrics, i.e., hit ratio (HR) and normalized discounted cumulative gain (NDCG). HR@k is defined as the fraction of cases that the ground-truth next item is among the top k items recommended, which emphasizes the accuracy of the model. NDCG@k is a position-aware metric, which emphasizes the rank of items, i.e., the top-ranked items are more important. Note that we empirically set k to 10, since the user always expects to find the target item from the first few items in the list.

5.1.3 Baseline Methods. To showcase the effectiveness of our MBASR, we integrate it into a wide range of representative models, including some RNN-based, attention-based and GNN-based models for multi-behavior sequential recommendation.

We consider the following six models:

- **RLBL** [18]. An RNN-based model that combines RNN and LBL (log-bilinear) to capture long-term and short-term preferences.
- **RIB** [45]. An RNN-based model that concatenates the item embedding and the behavior embedding as the input of a GRU layer. It then adopts an attention layer to distinguish the effects of different types of behaviors.
- **BINN** [16]. An RNN-based model that designs a contextual long short-term memory (CLSTM) network to integrate users' historical and current preferences.
- **GRUBAR** [7]. An extension model of BAR with GRU4Rec [10] as its backbone model. It uses a behavior attention layer to encode the relationship between the target behavior and items with various behaviors.
- **SASBAR** [7]. An extension model that utilizes BAR with SASRec [14] as the backbone model, where SASRec is an attention-based model that uses the multi-head attention mechanism to learn sequence representations.
- **GPG4HSR** [2]. A GNN-based model that constructs a global graph to capture the transitions between different types of behaviors, and utilizes a personalized graph to enhance the sequence representation with context information.

Moreover, we compare the boosted performance with the very state-of-the-art data augmentation method **RSS** [24], as it is a data-oriented augmentation method like our MBASR, which guarantees fairness in the experiments. It leverages a recency-based sampling strategy to generate new training samples such that every item in a sequence can be selected as the target item.

5.1.4 Implementation Details. We implement all the baseline methods and our MBASR using TensorFlow. For fair comparison, we use the Adam optimizer to train the model with the learning rate of 1e-3 and get the best model via early stopping w.r.t. HR@10 on the validation set. We set the sequence length L to 50, the batch size to 128, the dropout rate to 0.5, and the embedding dimension d to 64 for all methods. For the baselines, all hyper-parameters either follow the suggestions from the original papers or are carefully tuned, and the best results are reported. For RSS, we set the recency parameter α to 0.8. For our MBASR, we set the weighting coefficient λ to 0.8, and tune α, β and γ within the range of $\{0.2, 0.4, 0.6, 0.8\}$.

5.2 Overall Performance Comparison (RQ1)

We report the main experimental results in Table 2 and Table 3. Since each data augmentation operation is independent, in Table 2, we integrate our MBASR with the pairwise swapping (SS) data augmentation operation into various multi-behavior sequential recommendation models, and the performance of the other two augmentation operations will be demonstrated and discussed in RQ2. The best results are boldfaced, and "Imprv." indicates the improvement of our MBASR over the original model on HR@10. From Table 2, we can have the following key observations:

Table 3: Performance comparison of the data augmentation model RSS and our MBASR. Our MBASR is equipped with pairwise swapping as the data augmentation method. The best results are marked in bold, and the second-best results are underlined.

Model	Method	Tmall		UB		JD	
		HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
SASRec	-	<u>0.0473</u>	<u>0.0265</u>	0.0431	0.0224	0.3118	0.1674
	RSS	0.0431	0.0233	**0.0468**	<u>0.0244</u>	<u>0.3209</u>	<u>0.1684</u>
	MBASR	**0.0517**	**0.0278**	<u>0.0460</u>	**0.0246**	**0.3365**	**0.1782**
SASBAR	-	0.0493	<u>0.0283</u>	<u>0.0420</u>	0.0220	0.3085	<u>0.1652</u>
	RSS	<u>0.0500</u>	0.0281	0.0408	<u>0.0220</u>	<u>0.3105</u>	0.1613
	MBASR	**0.0539**	**0.0296**	**0.0494**	**0.0261**	**0.3217**	**0.1732**

- Our MBASR yields significant improvement to a wide range of mainstream MBSR models on all datasets, which demonstrates the superiority and applicability of our MBASR. Specifically, on average of the performance on the three datasets, our MBASR improves RLBL, RIB, BINN, GRUBAR, SASBAR and GPG4HSR by 25.17%, 4.55%, 5.5%, 7.62%, 10.20% and 4.18% in terms of HR@10, respectively. On the one hand, we view the users' historical interaction sequences from a more fine-grained perspective, and construct simple but informative samples through specific behavior-aware data augmentation operations. This facilitates the model in capturing additional sequential patterns and enhances its robustness and generalization capability. On the other hand, we mitigate the impact of noise on model performance using a position-based sampling strategy. The combination of these two ultimately brings encouraging performance.
- GPG4HSR emerges as the most competitive model among all the MBSR models, implying that the combination of global and local graphs can enhance the learning of relationships between behaviors and facilitate a better understanding of user intentions.
- SASBAR outperforms GRUBAR on all datasets, which suggests that the self-attention mechanism has advantages in modeling multi-behavior sequences.

To show the superiority of our MBASR, we compare it with the state-of-the-art data augmentation model, i.e., RSS. Specifically, we use SASRec and SASBAR as the backbone models, evaluating the performance improvement brought by RSS and our MBASR from the perspectives of single-behavior sequential recommendation and multi-behavior sequential recommendation, respectively. The results are presented in Table 3, where the best results are boldfaced and the second-best results are underlined. From Table 3, we can have the following key observations:

- Our MBASR consistently beats RSS on almost all datasets, and shows significant performance improvement in both representative SBSR and MBSR models. Such results generally demonstrate the superiority and applicability of our MBASR again. Compared with the original models, our MBASR offers the advantage of learning more enriched sequence representations from the constructed new samples, which is a crucial factor contributing significantly to boosting the model's performance.
- It can be observed that SASRec with RSS shows a great improvement on both UB and JD. However, when RSS is combined with the MBSR model SASBAR, it performs poorly in most cases. Such findings indicate that RSS does not adapt well to MBSR, as it fails to consider the impact of different types of behaviors on user preferences. At the same time, it further highlights the advantages of our MBASR in addressing the sparsity issue in multi-behavior sequences.
- SASRec outperforms SASBAR in most cases, and we believe that this phenomenon arises from the fact that we have sparsified the data, and models with simpler network structures tend to achieve better results when the data is sparse.

5.3 Effect of Augmentation Operators (RQ2)

To comprehensively validate the contributions of different data augmentation operations, we conduct a more detailed set of experiments. In Figure 2, we illustrate the performance of models when combining two data augmentation operations, namely, order perturbation and redundancy reduction, with various MBSR models across three datasets. As can be seen, the results exhibit a trend similar to that observed in RQ1, further reinforcing our findings. Each proposed data augmentation method brings varying degrees of performance improvement in all downstream tasks on each dataset. Taking Tmall as a case study, order perturbation improves RLBL, RIB, BINN, GRUBAR, SASBAR, and GPGHSR by 21.01%, 7.04%, 11.07%, 10.57%, 9.33%, and 7.49% in terms of HR@10, respectively. Correspondingly, for redundancy reduction, improvements on RLBL, RIB, BINN, GRUBAR, SASBAR, and GPGHSR are noted as 24.37%, 2.05%, 9.69%, 2.48%, 4.26%, and 5.87%, respectively, in terms of HR@10. This is another demonstration of the generality of our MBASR, and shows the design rationality of each data augmentation operation.

Notably, each augmentation method aligns with real shopping patterns, reinforcing the validity of our design principles. These experiments deepen our understanding of the effectiveness of our MBASR, providing compelling evidence for its feasibility in practical applications.

5.4 Effect of Data Sparsity (RQ3)

To investigate the impact of data sparsity on our MBASR, we intentionally inject sparsity into UB by extracting items from each user sequence at varying intervals, specifically 1, 3, and 5. This process resulted in three sparsified datasets, denoted as UB(1), UB(3), and UB(5), representing increasing levels of sparsity. In Figure 3, we show the HR@10 results of our MBASR with SASBAR as the backbone model when using the three different data augmentation operations on these three datasets.

The results align well with our expectation, which indicate a progressively poorer performance on HR@10 as the data sparsity

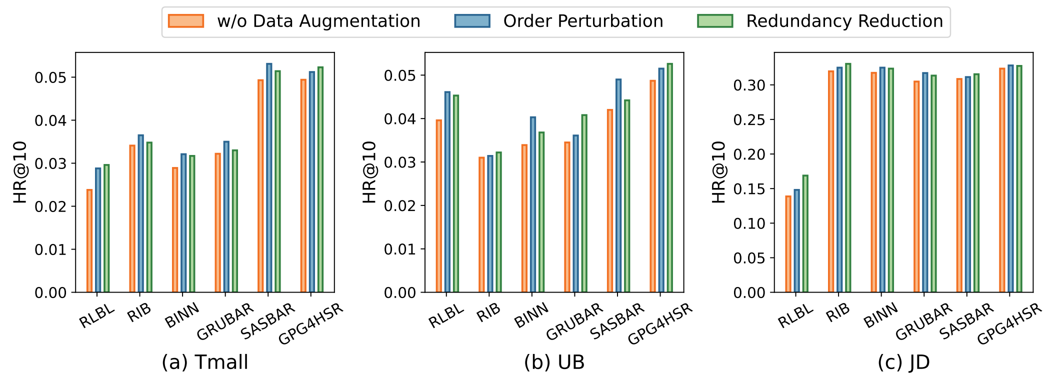

Figure 2: Experimental results of different MBSR models with our proposed data augmentation operations (order perturbation and redundancy reduction) or without (w/o) any data augmentation on the three datasets.

increases. In contrast, our MBASR can always enhance the original model's performance with its three data augmentation operations. This noteworthy observation indicates that our MBASR demonstrates superior capability when confronted with an increasingly sparse data.

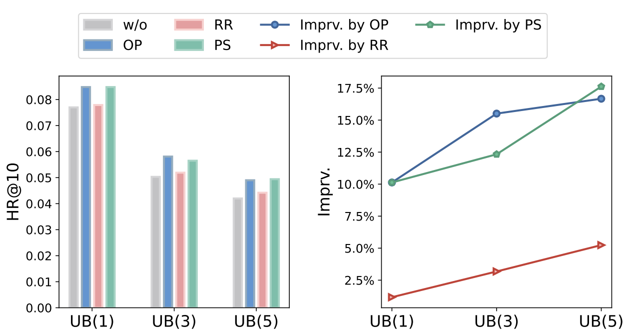

Figure 3: Impact of data sparsity on our MBASR. The left graph illustrates the performance of the model under three data augmentation operations as the degree of data sparsity increases. The right graph depicts the corresponding percentage improvement in performance.

5.5 Training Efficiency (RQ4)

Finally, to visualize the model performance, we present the performance curves of our MBASR with SASBAR as the backbone model and SASBAR over 100 epochs. All experiments are conducted on a single NVIDIA GeForce RTX 3090 and Figure 4 presents the results on UB for HR@10 and NDCG@10. The dash lines represent the performance without our MBASR. We can observe that around the first 10 epochs, the performance curves of the model using our MBASR and the original model almost overlap, indicating similar convergence speeds during this period. Subsequently, the performance of our MBASR continues to improve, while the performance of the original model stabilizes. Based on this observation, we conclude that, compared with the original model, our MBASR exhibits superior generalization capability. We attribute this improvement to the data augmentation operations we designed, which introduce some noise and diversity to the data, making the model more adaptable to real-world changes and uncertainties.

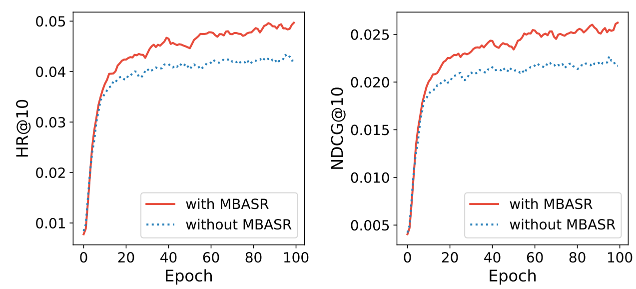

Figure 4: Training curve on UB.

6 CONCLUSION

In this paper, we study the problem of data sparsity in multi-behavior sequential recommendation. We propose a generic data-oriented framework called MBASR that contains three behavior-aware data augmentation methods (order perturbation, redundancy reduction, and pairwise swapping). We exploit the purchase behavior to divide an original sequence into multiple subsequences. Then, we employ tailored behavior-aware augmentation operations on these subsequences to simulate intricate user behaviors in real-world scenarios, thereby enriching the sequential patterns in user-item interactions. To mitigate noise introduced by randomness, we introduce a position-based sampling strategy, which can reduce the level of perturbations and enable a trade-off between enriching the samples and preserving the original information. In addition, we analyze the performance improvement brought by each augmentation method in different downstream tasks. Our approach is validated on three public datasets. Extensive experiments show that our MBASR can be seamlessly integrated into various multi-behavior sequential recommendation models, to improve their performance significantly. Besides, it can achieve better performance than the very state-of-the-art solution.

7 ACKNOWLEDGEME

We thank Miss Shu Chen for her helpful discussions, and the support of Guangdong Basic and Applied Basic Research Foundation (No. 2024A1515010122) and National Natural Science Foundation of China (Nos. 62272315 and 62172283).

REFERENCES

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20, pages 1571–1607, 2020.

[2] Weixin Chen, Mingkai He, Yongxin Ni, Weike Pan, Li Chen, and Zhong Ming. Global and personalized graphs for heterogeneous sequential recommendation by learning behavior transitions and user intentions. In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys '22, pages 268–277, 2022.

[3] Junsu Cho, Dongmin Hyun, Dong won Lim, Hyeon jae Cheon, Hyoung-iel Park, and Hwanjo Yu. Dynamic multi-behavior sequence modeling for next item recommendation. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, AAAI '23, pages 4199–4207, 2023.

[4] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. Sequential user-based recurrent neural network recommendations. In *Proceedings of the 11th ACM Conference on Recommender Systems*, RecSys '17, pages 152–160, 2017.

[5] Ziwei Fan, Ke Xu, Zhang Dong, Hao Peng, Jiawei Zhang, and Philip S. Yu. Graph collaborative signals denoising and augmentation for recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, pages 2037–2041, 2023.

[6] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent a new approach to self-supervised learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NeurIPS '20, pages 21271–21284, 2020.

[7] Mingkai He, Weike Pan, and Zhong Ming. BAR: Behavior-aware recommendation for sequential heterogeneous one-class collaborative filtering. *Information Sciences*, 608(C):881–899, 2022.

[8] Ruining He and Julian McAuley. Fusing similarity models with Markov chains for sparse sequential recommendation. In *Proceedings of the 16th IEEE International Conference on Data Mining*, ICDM '16, pages 191–200, 2016.

[9] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 173–182, 2017.

[10] Balázs Hidasi and Alexandros Karatzoglou. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pages 843–852, 2018.

[11] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *Proceedings of the 4th International Conference on Learning Representations*, ICLR '16, 2016.

[12] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining*, ICDM '08, pages 263–272, 2008.

[13] Juyong Jiang, Yingtao Luo, Jae Boum Kim, Kai Zhang, and Sunghun Kim. Sequential recommendation with bidirectional chronological augmentation of transformer. *arXiv preprint arXiv:2112.06460*, 2021.

[14] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *Proceedings of the 18th IEEE International Conference on Data Mining*, ICDM '18, pages 197–206, 2018.

[15] Ziyi Kou, Saurav Manchanda, Shih-Ting Lin, Min Xie, Haixun Wang, and Xiangliang Zhang. Modeling sequential collaborative user behaviors for seller-aware next basket recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, pages 1097–1106, 2023.

[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*, KDD '18, pages 1734–1743, 2018.

[17] Jing Lin, Weike Pan, and Zhong Ming. FISSA: Fusing item similarity models with self-attention networks for sequential recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, pages 130–139, 2020.

[18] 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.

[19] Yong Liu, Yingtai Xiao, Qiong Wu, Chunyan Miao, Juyong Zhang, Binqiang Zhao, and Haihong Tang. Diversified interactive recommendation with implicit feedback. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, AAAI '20, pages 4932–4939, 2020.

[20] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S Yu, Julian McAuley, and Caiming Xiong. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479*, 2021.

[21] Zhiwei Liu, Ziwei Fan, Yu Wang, and Philip S. Yu. Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pages 1608–1612, 2021.

[22] 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*, AAAI '20, pages 5045–5052, 2020.

[23] 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*, SIGIR '20, pages 1091–1100, 2020.

[24] Aleksandr Petrov and Craig Macdonald. Effective and efficient training for sequential recommendation using recency sampling. In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys '22, pages 81–91, 2022.

[25] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. GCC: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '20, pages 1150–1160, 2020.

[26] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, WSDM '22, pages 813–823, 2022.

[27] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi YIn. Rethinking the item order in session-based recommendation with graph neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, pages 579–588, 2019.

[28] Mingyue Qu, Nan Wang, and Jinbao Li. A two-path multibehavior model of user interaction. *Electronics*, 12(14), 2023.

[29] 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*, RecSys '17, pages 130–137, 2017.

[30] 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*, WWW '10, pages 811–820, 2010.

[31] Jiajie Su, Chaochao Chen, Zibin Lin, Xi Li, Weiming Liu, and Xiaolin Zheng. Personalized behavior-aware transformer for multi-behavior sequential recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, pages 6321–6331, 2023.

[32] 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*, WSDM '18, pages 565–573, 2018.

[33] Jianling Wang, Ya Le, Bo Chang, Yuyan Wang, Ed H. Chi, and Minmin Chen. Learning to augment for casual user recommendation. In *Proceedings of the 31th International Conference on World Wide Web*, WWW '22, page 2183–2194, 2022.

[34] 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 29th International Conference on World Wide Web*, WWW '20, pages 3056–3062, 2020.

[35] Zhenlei Wang, Jingsen Zhang, Hongteng Xu, Xu Chen, Yongfeng Zhang, Wayne Xin Zhao, and Ji-Rong Wen. Counterfactual data-augmented sequential recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pages 347–356, 2021.

[36] Zihan Wei, Ning Wu, Fengxia Li, Ke Wang, and Wei Zhang. MoCo4SRec: A momentum contrastive learning framework for sequential recommendation. *Expert Systems with Applications*, 223:119911, 2023.

[37] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pages 726–735, 2021.

[38] 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*, AAAI '19, pages 346–353, 2019.

[39] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *2022 IEEE 38th International Conference on Data Engineering*, ICDE '22, pages 1259–1273, 2022.

[40] 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 28th International Joint Conference on Artificial Intelligence*, IJCAI '19, pages 3940–3946, 2019.

[41] Bo Yu, Ruoqian Zhang, Wei Chen, and Junhua Fang. Graph neural network based model for multi-behavior session-based recommendation. *Geoinformatica*, 26(2):429–447, 2022.

[42] Enming Yuan, Wei Guo, Zhicheng He, Huifeng Guo, Chengkai Liu, and Ruiming Tang. Multi-behavior sequential transformer recommender. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, pages 1642–1652, 2022.

[43] Zhuoxin Zhan, Mingkai He, Weike Pan, and Zhong Ming. Transrec++: Translation-based sequential recommendation with heterogeneous feedback. *Frontiers Comput. Sci.*, 16(2):162615, 2022.

[44] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, CIKM '20, pages 1893–1902, 2020.

[45] 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*, WSDM '18, pages 727–735, 2018.