Augmenting Sequential Recommendation with Balanced Relevance and Diversity

Yizhou Dang [1], Jiahui Zhang [1], Yuting Liu [1], Enneng Yang [1], Yuliang Liang [1]
Guibing Guo [1*], Jianzhe Zhao [1*], Xingwei Wang [2]
[1] Software College, Northeastern University, China
[2] School of Computer Science and Engineering, Northeastern University, China
{yzdang,yutingliu,ennengyang,liangyuliang}@stumail.neu.edu.cn
{guogb,zhaojz}@swc.neu.edu.cn, {zhangjiahui,wangxw}@mail.neu.edu.cn

Abstract

By generating new yet effective data, data augmentation has become a promising method to mitigate the data sparsity problem in sequential recommendation. Existing works focus on augmenting the original data but rarely explore the issue of imbalanced relevance and diversity for augmented data, leading to semantic drift problems or limited performance improvements. In this paper, we propose a novel Balanced data Augmentation Plugin for Sequential Recommendation (BASRec) to generate data that balance relevance and diversity. BASRec consists of two modules: Single-sequence Augmentation and Cross-sequence Augmentation. The former leverages the randomness of the heuristic operators to generate diverse sequences for a single user, after which the diverse and the original sequences are fused at the representation level to obtain relevance. Further, we devise a reweighting strategy to enable the model to learn the preferences based on the two properties adaptively. The Cross-sequence Augmentation performs nonlinear mixing between different sequence representations from two directions. It produces virtual sequence representations that are diverse enough but retain the vital semantics of the original sequences. These two modules enhance the model to discover fine-grained preferences knowledge from single-user and cross-user perspectives. Extensive experiments verify the effectiveness of BASRec. The average improvement is up to 72.0% on GRU4Rec, 33.8% on SASRec, and 68.5% on FMLP-Rec. We demonstrate that BASRec generates data with a better balance between relevance and diversity than existing methods. The source code is available at https://github.com/KingGugu/BASRec.

Introduction

As an essential branch of recommender systems, sequential recommendation (SR) has received much attention due to its well-consistency with real-world recommendation situations. However, the widespread problem of data sparsity limits the SR model's performance (Jing et al. 2023). For this reason, researchers have proposed many data augmentation methods to mitigate this phenomenon (Dang et al. 2023b, 2024a). Earlier work used heuristic methods to directly augment sequences and mix them to the training process, such as Sliding Windows (Tang and Wang 2018) and Dropout (Tan, Xu, and Liu 2016). Later, some researchers generated high-quality augmented data by counterfactual thinking (Wang et al. 2021), diffusion models (Liu et al. 2023) or bi-directional transformer (Jiang et al. 2021). With the success of self-supervised learning (Chen et al. 2020), many sequence-level augmentation operators have been proposed for contrastive learning (Liu et al. 2021; Xie et al. 2022).

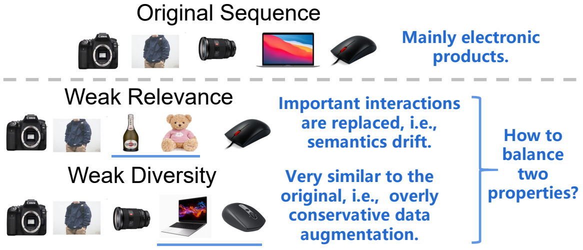
Original Sequence
Mainly electronic products.
Weak Relevance
Important interactions are replaced, i.e., semantics drift.
Weak Diversity
Very similar to the original, i.e., overly conservative data augmentation.
How to balance two properties?

Figure 1: An illustration of imbalanced relevance and diversity issues in sequential data augmentation.

Despite the effectiveness, the imbalance between relevance and diversity for augmented data remains to be solved (Bian et al. 2022). 'Relevance' means that the augmented data should have transition patterns similar to the original data to avoid semantics drift problems. 'Diversity' means that the augmented data should contain sufficient variations to enable the model to explore more user preferences knowledge and improve its performance. However, these two factors are often conflicting and challenging to trade off. For example, as shown in Figure 1, heuristic augmentation and operators proposed in recent years perform cropping, masking, substitution, or shuffle to the original sequence. The new sequences may deviate from the original data, resulting in weak correlations. Model- and representation-level augmentation methods, while improving the correlation of the generated data through well-designed generation modules, fail to produce sufficient diversity. These conservative augmented data make it challenging to improve the model performance further. Intuitively, we can obtain suitable samples by directly merging the two augmented sequences in Figure 1. However, directly merging them will corrupt the original sequence patterns, and irrelevant or repeated items will be retained, resulting in inaccurate representation learning. In addition, existing methods focus on single-sequence

[*] Corresponding Authors.

augmentation and lack consideration for the cross-sequence preference patterns and semantic information.

In this paper, inspired by previous work based on mixup (Zhang et al. 2018; Bian et al. 2022), we propose BASRec, Balanced data Augmentation plugin for Sequential Recommendation. Our core idea is to generate new samples that balance relevance and diversity through representation-level fusion. The BASRec consists of two modules: Single-sequence Augmentation and Cross-sequence Augmentation. For Single-sequence Augmentation, we propose two new mixup-based data augmentation operators, M-Substitute and M-Reorder. On top of the traditional operator, we improve the diversity by sampling the operation weights from a uniform distribution, while the correlation is obtained by fusing the original and the augmented sequence representation. Further, we adaptively reweight the training loss with the operators' augmentation weights and mixup weights, allowing the model to learn based on the difference between the augmented and the original samples. So far, all augmentations are limited to a single sequence. For Cross-sequence Augmentation, we propose to explore the semantics of sequence and preferences knowledge among different users. The traditional mixup operation can only generate samples in the linear dimension for a pair of samples (Guo 2020). In order to further improve the space of synthetic samples and enable the model to discover the fine-grained knowledge among different users, we adopt a nonlinear mixup approach for fusion. However, assigning a weight to each parameter introduces a significant computational overhead. We decompose this process and further present the Item-wise and Feature-wise mixup. Our method generates new training samples in the representation space that relate to the original data but contain diverse transition patterns, balancing the two properties. Finally, we employ a two-stage learning strategy to avoid the noise and convergence instability introduced by the hybrid representation at the beginning of training.

Extensive experiments on four real-world datasets with four base SR models demonstrate that our proposed BASRec achieves significant improvements. We compare BASRec with heuristic and training-required data augmentation methods to further validate its superiority. Besides, We experimentally demonstrate that the data generated by BASRec strikes a better balance between the two properties. The main contributions can be summarized follows:

- We emphasize the unbalanced relevance and diversity of current data augmentation methods and propose a Balanced Data Augmentation Plugin for Sequential Recommendation, which can be seamlessly integrated into most existing sequential recommendation models.
- We design two key modules, Single-sequence Augmentation and Cross-sequence Augmentation. They perform augmentation and fusion operations to synthesize new samples that balance relevance and diversity.
- We conduct comprehensive experiments on real-world datasets to demonstrate the effectiveness of BASRec, showing significant improvements over various sequential models. Our approach also achieves competitive performance compared to the data augmentation baselines.

Related Work

Sequential Recommendation

Sequential recommendation aims to provide users with personalized suggestions based on their historical behaviors. Early works leveraged Markov Chains (Rendle, Freudenthaler, and Schmidt-Thieme 2010) and session-based KNN (He and McAuley 2016; Hu et al. 2020) to model sequential data. Later, researchers adopted CNNs (Tang and Wang 2018) and RNNs (Liu et al. 2016) to capture the relationships among items. For instance, NextItNet (Yuan et al. 2019) combined masked filters with 1D dilated convolutions to model the sequential dependencies. GRU4Rec (Hidasi et al. 2015) used gated recurrent units to capture behavioral patterns. More recently, Transformers (Vaswani et al. 2017) showed extraordinary performance in learning item importance and behavior relevance for next-item prediction. SASRec (Kang and McAuley 2018) is the representative work that adopted the multi-head attention mechanism to perform sequential recommendation. BERT4Rec (Sun et al. 2019) introduced a cloze task with a bidirectional attentive encoder. To model dynamic uncertainty and capture collaborative transitivity, STOSA (Fan et al. 2022) embeds each item as a stochastic Gaussian distribution and devises a Wasserstein Self-Attention module to characterize item-item relationships. In addition, cross-domain (Zhao et al. 2023), multi-modal (Zhang, Zhou, and Shen 2023), and multi-behavior (Su et al. 2023) SR tasks have also been widely explored.

Data Augmentation for SR

Data augmentation has been widely used in many domains to improve model performance and robustness (Dang et al. 2024b). It also received extensive attention in sequential recommendation. Slide Windows (Tang and Wang 2018) and Dropout (Tan, Xu, and Liu 2016) are pioneer works that split one sequence into many sub-sequences or discard some items from the original data. Later, many training-required data synthesis methods are proposed since heuristics may produce low-quality augmented data. For example, DiffASR (Liu et al. 2023) adopted the diffusion model for sequence generation. Two guide strategies are designed to control the model to generate the items corresponding to the raw data. CASR (Wang et al. 2021) proposed substituting some previously purchased items with other unknown items based on counterfactual thinking. L2Aug (Wang et al. 2022) enhanced casual users' sequences by learning from core users' interaction patterns. Besides, some work has explored using self-supervised signals to mitigate data sparsity in SR. CL4SRec (Xie et al. 2022) constructed contrastive views by three augmentation operators, i.e., item crop, mask, and reorder. DuoRec (Qiu et al. 2022) explored the representation degeneration issue in SR and solved it by contrastive regularization. ReDA (Bian et al. 2022) adopted a neural retriever to retrieve augmentation users and conducted two types of representation augmentation.

Unlike these studies, we propose a Balanced Data Augmentation Plugin to generate data that considers relevance and diversity. It performs augmentation by elaborate augmentation and mixup operations in the representation space.

Preliminaries

Problem Formulation

Suppose we have user and item sets denoted by $\mathcal{U}$ and $\mathcal{V}$, respectively. Each user $u \in \mathcal{U}$ is associated with a sequence of interacted items in chronological order $s_u = [v_1, v_2, \ldots, v_{|s_u|}]$, where $v_j \in \mathcal{V}$ indicate the item that user u has interacted with at time step j. The $|s_u|$ is the sequence length. Given the sequences of interacted items s_u, SR aims to accurately predict the possible item v^* that user u will interact with at time step $|s_u| + 1$, formulated as follows:

$$\underset{v^* \in \mathcal{V}}{\arg\max} \; P\left(v_{|s_u|+1} = v^* \mid s_u\right). \tag{1}$$

The model will calculate the probability of all candidate items and select the highest one for recommendation.

Mixup for Data Augmentation

Mixup (Zhang et al. 2018; Zhang, Yu, and Zhang 2020) is a simple yet effective data augmentation method. It implements linear interpolation in the input space to construct virtual training data. Given two input samples x_i, x_j along with the labels y_i, y_j, the mixup process can be formulated as:

$$\tilde{x} = \lambda \cdot x_i + (1 - \lambda) \cdot x_j, \tag{2}$$

$$\tilde{y} = \lambda \cdot y_i + (1 - \lambda) \cdot y_j. \tag{3}$$

where $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$ is the mixup coefficient from beta distribution. Some work in the recommendation field has explored the use of mixup to synthesize hard negative samples for training graph neural networks better (Huang et al. 2021) or to improve the representation of tail sessions (Yang et al. 2023a). However, these efforts failed to balance relevance and diversity when performing augmentation. They may generate harmful or overly conservative augmented data. Also, they are limited by the type of backbone network and have lower generalization capability (Bian et al. 2022).

Ours: BASRec

In this section, we present our proposed BASRec. The overall framework is illustrated in Figure 2. Our approach consists of two separate modules: Single-sequence Augmentation and Cross-sequence Augmentation. After that, we introduce the training and inference process of BASRec. Finally, we provide a discussion about our and existing methods.

Single-sequence Augmentation

Single-sequence Augmentation generates new samples by mixing the representations of items in the original sequence with those in the augmented sequence. Specifically, we propose two new operators, M-Reorder and M-Substitute, to accomplish this augmentation operation.

The standard SR paradigm (Kang and McAuley 2018; Bian et al. 2022) maintains an item embedding matrix $\mathbf{M}_{\mathcal{V}} \in \mathbb{R}^{|\mathcal{V}| \times D}$. The matrix project the high-dimensional one-hot representation of an item to low-dimensional dense representations. Given an original user sequence $s_u = [v_1, v_2, \ldots, v_{|s_u|}]$, the Look-up operation will be applied for $\mathbf{M}_{\mathcal{V}}$ to get a sequence of item representation, i.e., $E_u = [m_{v_1}, m_{v_2}, \ldots, m_{v_{|s_u|}}]$. Note that we omit the padding for short sequences and intercepting for long sequences.

M-Reorder. Given an original sequence s_u, M-Reorder first selects a sub-sequence with length $c = rate \cdot |s_u|$. Unlike traditional operators that preform augmentation with a fixed $rate$, we further extend the augmentation possibilities by drawing $rate$ from a uniform distribution:

$$rate \sim \mathrm{Uniform}(a, b), \tag{4}$$

where a and b are hyper-parameters and $0 < a < b < 1$. Then, we randomly shuffle this sub-sequence $[v_i, \cdots, v_{i+c-1}]$ as $\left[v'_i, \ldots, v'_{i+c-1}\right]$ and get the augmented item sequence s'_u:

$$s'_u = \mathrm{Reorder}\left(s_u\right) = \left[v_1, v_2, \cdots, v'_i, \cdots, v'_{i+r-1}, \cdots, v_n\right]. \tag{5}$$

Unlike traditional operators that directly use s'_u as a new sample to participate in model training, we mix up the sequence of item representation corresponding to s_u and s'_u to generate new training samples in the representation space:

$$E'_u = \text{Look-up}\left(\mathbf{M}_{\mathcal{V}}, s'_u\right), \tag{6}$$

$$E_u^{In} = \lambda \cdot E_u + (1 - \lambda) \cdot E'_u, \tag{7}$$

where $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$ is the mixup weight and E_u^{In} is augment representation used for model training.

M-Substitute. This operator is similar to M-Reorder. Given an original sequence s_u, it first randomly selects $c = rate \cdot |s_u|$ different indices $\{\mathrm{idx}_1, \mathrm{idx}_2, \ldots, \mathrm{idx}_c\}$, where $rate$ is sampled following Eq. 4. Then, we replace each with a correlated item based on the selected indices. We adopt the cosine similarity method as (Liu et al. 2021) to select the items to substitute. The above process can be formulated as:

$$s'_u = \mathrm{Substitute}\left(s_u\right) = \left[v_1, v_2, \ldots, \bar{v}_{\mathrm{idx}_i}, \ldots, v_{|s_u|}\right]. \tag{8}$$

Following the same steps as in M-Reorder, through Eq. 6 and Eq. 7, we can obtain the augmented representation E_u^{In}.

Adaptive Loss Weighting. The augmentation of the original representation by the two operators comes from two main dimensions: 1) The change of the operator to the original interaction sequence, i.e., the $rate$ of operators. 2) The mixing of the new sequence representation with the original one, i.e., the λ of mixup. To further measure this augmentation process so that the model can adaptively learn the preferences based on the relevance and diversity of the augmentation, we propose an adaptive loss weighting strategy. Inspired by previous work (Yang et al. 2023b), based on the $rate$ from two operators and mixup coefficient λ from $\mathrm{Beta}(\alpha, \alpha)$, we define the transformation as follows:

$$\omega^{(1)} = 1/\left(rate \cdot \lambda\right); \; \omega^{(2)} = \frac{\omega^{(1)} - \omega^{(1)}_{\min}}{\omega^{(1)}_{\max} - \omega^{(1)}_{\min}}. \tag{9}$$

We adopt $\omega = \omega^{(2)}$ as the output. For each augmented representation E_u^f, we assign it with an exclusive weight. This weighting process also guides the model in distinguishing how much of the original representation has been injected with the new representation, further improving the robustness of the model. This weight will be used in model training, and we will introduce it in the Model Training section.

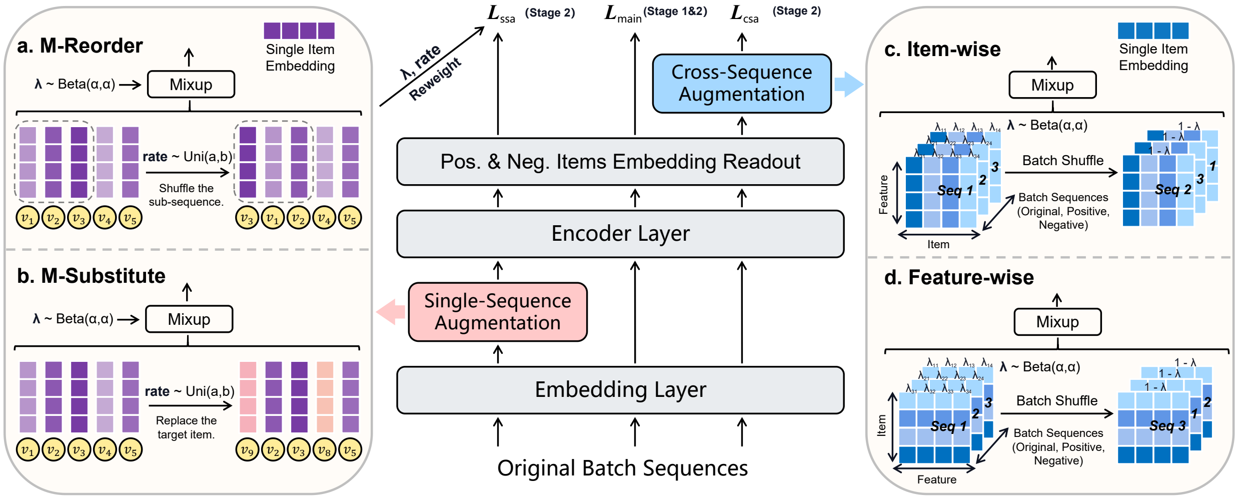

Figure 2: The overview of the proposed Balanced data Augmentation method.

Cross-sequence Augmentation

In Single-sequence Augmentation, the newly generated samples are limited to only independent single users. In recommender systems, each user is likely to have overlapping behaviors and preferences with other users, i.e., collaboration information (Luo, Liu, and Pan 2024; Cheng et al. 2024). Therefore, for Cross-sequence Augmentation, we generate new samples by mixing the different sequence output to discover the preference knowledge among different users further. By feeding the sequence representation E_u and E_u^{In} into the Encoder, we can obtain the output representation of the sequence H_u and H_u^{In}. In sequential recommendation, user representations are usually modeled implicitly, so H_u and H_u^{In} can also be interpreted as user representations. Note that the two modules we proposed are independent augmentation lines. The representation of a sequence that has been augmented by Single-sequence Augmentation will not be fed into Cross-sequence Augmentation.

The base mixup can only create samples in linear space. Inspired by the nonlinear mixup (Guo 2020), we attempt to assign different weights to every parameter in H_u. However, this mixup strategy incurs significant computational overhead. Thus, we further decompose this process into Item-wise mixup and Feature-wise mixup. Given a batch of sequences $\{s_u\}_{u=1}^{B}$, we can obtain the corresponding batch of representations $\{H_u\}_{u=1}^{B} \in R^{B\times N\times D}$, where B, N and D are the batch size, maximum sequence length, and embedding dimensions, respectively. We start by shuffling $\{H_u\}_{u=1}^{B}$ from the batch perspective:

$$\{H'_u\}_{u=1}^{B} = \text{Shuffle}\,\{H_u\}_{u=1}^{B} \,. \tag{10}$$

For **Item-wise Nonlinear Mixup**, we sample a mixup weights matrix $\Lambda_I \in R^{B\times N}$ from the $\text{Beta}(\alpha,\alpha)$, where N is the predefined maximum sequence length. The mixup is performed as follows:

$$\{H_u^{Out}\}_{u=1}^{B} = \Lambda_I \circ \{H_u\}_{u=1}^{B} + (1-\Lambda_I) \circ \{H'_u\}_{u=1}^{B} \,, \tag{11}$$

where $\circ$ denotes the Hadamard product. For **Feature-wise Nonlinear Mixup**, we sample a mixup weights matrix $\Lambda_F \in R^{B\times D}$ from the $\text{Beta}(\alpha,\alpha)$ and perform similar operation:

$$\{H_u^{Out}\}_{u=1}^{B} = \Lambda_F \circ \{H_u\}_{u=1}^{B} + (1-\Lambda_F) \circ \{H'_u\}_{u=1}^{B} \,. \tag{12}$$

The Cross-sequence Augmentation will simultaneously perform the same mixup process for the representations of positive and negative items. The final output is used to calculate the recommendation loss. The above two processes can be executed multiple times, and each mixing produces new virtual representations in the representation space.

Model Training

Following previous works (Kang and McAuley 2018; Dang et al. 2023a), we adopt the commonly used binary cross-entropy (BCE) loss for the sequential recommendation task:

$$\begin{aligned}\mathcal{L}_{main} &= \text{BCE}\left(H_u, E_u^{+}, E_u^{-}\right)\\ &= -\left[\log\left(\sigma\left(H_u \cdot E_u^{+}\right)\right) + \log\left(1-\sigma\left(H_u \cdot E_u^{-}\right)\right)\right],\end{aligned} \tag{13}$$

where H_u, E_u^{+} and E_u^{-} denote the representations of the user, the positive and negative items, respectively. $\sigma()$ is the sigmoid function. The representations involved in the $\mathcal{L}_{main}$ will not be augmented. For each sequence, we apply two operators for augmentation, respectively. The loss is calculated based on H_u^{In}, original positive and negative item representations. Further, we use the pre-computed weights ω to reweight this objective function:

$$\mathcal{L}_{ssa} = \omega \cdot \text{BCE}\left(H_u^{In}, E_u^{+}, E_u^{-}\right). \tag{14}$$

For Cross-sequence Augmentation, we use all mixed representations to calculate the recommendation loss:

$$\mathcal{L}_{csa} = \text{BCE}\left(H_u^{Out}, E_u^{Out+}, E_u^{Out-}\right). \tag{15}$$

Mixing representations at the beginning of training with Randomly initialized embedding may introduce noise and convergence problems. To tackle this, we adopt a two-stage training strategy. In the first stage, we follow the standard sequential recommendation model training process, with the primary objective of facilitating the learning of high-quality representations for items. We only use Eq. 13 as the objective function at this stage. In the second stage, we employ the two augmentation modules as described previously:

$$\mathcal{L} = \mathcal{L}_{main} + \mathcal{L}_{ssa} + \mathcal{L}_{csa}. \tag{16}$$

We do not set additional loss weights for $\mathcal{L}_{ssa}$ and $\mathcal{L}_{csa}$ for the following reasons: 1) The $\mathcal{L}_{ssa}$ has been reweighted in the previous sections. 2) For the two proposed modules, we treat the data augmented by them equally to the original data during the training process. Each augmented data can be considered as a new user and interaction generated in the representation space. During the inference phase, all augmentation modules are deactivated.

Discussion and Analysis

Comparison with Existing Methods. For Single-sequence Augmentation, sampling $rate$ from a uniform distribution improves the diversity of the augmented sequence. The fusion with the original sequence representation and the loss reweighting process enables the model to learn based on the correlation of the augmented samples with the original samples. For Cross-sequence Augmentation, our cross-user nonlinear mixup strategy endows the learning process with diverse but relevant preference knowledge and collaborative signals among different users. Traditional operators (Liu et al. 2021; Xie et al. 2022) that directly edit the original sequence with a fixed $rate$ may lead to problems of conservative data augmentation or semantics drift. Editing the original sequence may also remove critical interactions (Dang et al. 2023b). Some work craft trainable data generators to augment the data (Wang et al. 2021; Liu et al. 2023; Wang et al. 2022). However, these methods can only produce discrete user interactions while introducing additional learnable parameters. Our method can generate more samples at the sequence level or across sequences in the representation space without additional model parameters. Besides, some approaches are also limited by the type of backbone network (Jiang et al. 2021; Bian et al. 2022), whereas BASRec is a model-agnostic augmentation plugin.

Complexity Analysis. We choose SASRec as the backbone model for explanation. Other choices can be analyzed similarly. Since Our BASRec does not introduce any auxiliary learnable parameters, the model size of BASRec is identical to SASRec. The time complexity of SASRec is mainly due to the self-attention module, which is $O\left(N^2 D|\mathcal{U}|\right)$ (Xie et al. 2022). The time complexity for calculating loss is $\mathcal{O}\left(ND|\mathcal{U}|\right)$. Considering our method, for two operators in BASRec, the complexity of operation is $O\left((a+b)N|\mathcal{U}|/2\right)$. Suppose we need to perform a total of Q mixup operations for each sequence, so the time complexity is $O\left(QD|\mathcal{U}|\right)$ since the mixing process is performed through Hadamard product. The total time complexity of SASRec and BASRec are $\mathcal{O}\left(\left(N^2+N\right)D|\mathcal{U}|\right)$ and $\mathcal{O}\left(\left(N^2+N+Q\right)D|\mathcal{U}|\right)$[1], respectively. Their analytical complexity is the same in magnitude. Our BASRec can generate data with acceptable additional time costs.

Dataset	**Beauty**	**Sports**	**Yelp**	**Home**
# Users	22,363	35,958	30,431	66,519
# Items	12,101	18,357	20,033	28,237
# Inter	198,502	296,337	316,354	551,682
# AvgLen	8.9	8.3	10.4	8.3
Sparsity	99.92%	99.95%	99.95%	99.97%

Table 1: The statistics of four datasets. The 'Inter' and 'AvgLen' denote the number of interactions and average length.

Experiments

Experimental Settings

Datasets. We adopt four widely-used public datasets: Beauty, Sports, and Home are obtained from Amazon (McAuley, Pandey, and Leskovec 2015) with user reviews of products. Yelp[2] is a business dataset. We use the transaction records after January 1st, 2019. Users/items with fewer than five interactions are filtered out (Liu et al. 2021). The detailed statistics are summarized in Table 1.

Baselines. The baselines consist of three categories. The first category is general models to validate the effectiveness of BASRec, including GRU4Rec (Hidasi et al. 2015), NextItNet (Yuan et al. 2019), SASRec (Kang and McAuley 2018) and FMLPRec (Zhou et al. 2022). These models employ diverse architectures, including RNN, CNN, Transformer, and MLP. The second category is heuristic augmentation methods: Random (Ran) and Random-seq (Ran-S) (Liu et al. 2023), Slide Windows (SW) (Tang and Wang 2018), CMR (Xie et al. 2022) and CMRSI (Liu et al. 2021). The third category is training-required augmentation models: ASReP (Jiang et al. 2021), DiffuASR (Liu et al. 2023), and CL4SRec (Xie et al. 2022). Details about baselines are provided in the Appendix. We do not include methods ReDA (Bian et al. 2022), CASR (Wang et al. 2021), and L2Aug (Wang et al. 2022) since they do not provide open-source codes for reliable reproduction (empty code repository or no available code repository links).

Implementation Details. For all baselines, we adopt the implementation provided by the authors. We set the embedding size to 64 and the batch size to 256. The maximum sequence length is set to 50. To ensure fair comparisons, we carefully set and tune all other hyper-parameters of each method as reported and suggested in the original papers. We use the Adam (Kingma and Ba 2014) optimizer with the learning rate 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$. For BASRec, we tune the α, a, b in the range of $\{0.2, 0.3, 0.4, 0.5, 0.6\}$, $\{0.1, 0.2, 0.3\}$, $\{0.6, 0.7, 0.8\}$, respectively. We conduct five runs and report the average results for all methods. Generally, *greater* values imply *better* ranking accuracy.

[1] The time cost of operators, $O\left((a+b)N|\mathcal{U}|/2\right)$, is omitted since it has a lower complexity order.

[2] https://www.yelp.com/dataset

Method	Beauty				Sports				Yelp				Home			
	N@10	H@10	N@20	H@20	N@10	H@10	N@20	H@20	N@10	H@10	N@20	H@20	N@10	H@10	N@20	H@20
GRU4Rec	0.0208	0.0412	0.0273	0.0670	0.0069	0.0146	0.0101	0.0274	0.0084	0.0174	0.0121	0.0325	0.0032	0.0066	0.0046	0.0123
w\ Ours	**0.0286**	**0.0546**	**0.0360**	**0.0842**	**0.0164**	**0.0307**	**0.0208**	**0.0483**	**0.0140**	**0.0289**	**0.0194**	**0.0502**	**0.0063**	**0.0128**	**0.0084**	**0.0214**
Improve	37.50%	32.52%	31.87%	25.67%	137.68%	110.27%	105.94%	76.28%	66.67%	66.09%	60.33%	54.46%	96.88%	93.94%	82.61%	73.98%
NextItNet	0.0163	0.0326	0.0210	0.0511	0.0077	0.0154	0.0106	0.0272	0.0109	0.0222	0.0155	0.0406	0.0033	0.0070	0.0046	0.0125
w\ Ours	**0.0202**	**0.0365**	**0.0253**	**0.0589**	**0.0091**	**0.0193**	**0.0128**	**0.0320**	**0.0134**	**0.0282**	**0.0183**	**0.0465**	**0.0040**	**0.0087**	**0.0063**	**0.0162**
Improve	23.93%	11.96%	20.48%	15.26%	18.18%	25.32%	20.75%	17.65%	22.94%	27.03%	18.06%	14.53%	21.21%	24.29%	36.96%	29.60%
SASRec	0.0338	0.0639	0.0413	0.0935	0.0174	0.0320	0.0214	0.0482	0.0136	0.0277	0.0180	0.0453	0.0078	0.0149	0.0100	0.0239
w\ Ours	**0.0455**	**0.0810**	**0.0539**	**0.1145**	**0.0242**	**0.0436**	**0.0294**	**0.0641**	**0.0164**	**0.0326**	**0.0216**	**0.0537**	**0.0128**	**0.0223**	**0.0154**	**0.0327**
Improve	34.62%	26.76%	30.51%	22.46%	39.08%	36.25%	37.38%	32.99%	20.59%	17.69%	20.00%	18.54%	64.10%	49.66%	54.00%	36.82%
FMLP-Rec	0.0298	0.0563	0.0361	0.0814	0.0131	0.0255	0.0163	0.0383	0.0093	0.0195	0.0134	0.0357	0.0071	0.0134	0.0091	0.0215
w\ Ours	**0.0441**	**0.0767**	**0.0519**	**0.1076**	**0.0240**	**0.0432**	**0.0286**	**0.0615**	**0.0180**	**0.0347**	**0.0233**	**0.0559**	**0.0147**	**0.0245**	**0.0174**	**0.0353**
Improve	47.99%	36.23%	43.77%	32.19%	83.21%	69.41%	75.46%	60.57%	93.55%	77.95%	73.88%	56.58%	107.04%	82.84%	91.21%	64.19%

Table 2: Performance comparison of four backbone models and BASRec on four datasets. The 'w/ Ours' represents adding our BASRec. All improvements are statistically significant, as determined by a paired t-test with $p \leq 0.05$.

BackBone	GRU4Rec								SASRec							
Method	Beauty		Sports		Yelp		Home		Beauty		Sports		Yelp		Home	
	N@10	H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10	H@10
Base	0.0208	0.0412	0.0069	0.0146	0.0084	0.0174	0.0032	0.0066	0.0338	0.0639	0.0174	0.0320	0.0136	0.0277	0.0078	0.0149
Ran	0.0212	0.0471	0.0083	0.0171	0.0087	0.0189	0.0036	0.0075	0.0285	0.0553	0.0186	0.0341	0.0162	0.0316	0.0086	0.0156
SW	0.0192	0.0501	0.0082	0.0159	0.0090	0.0195	0.0040	0.0083	0.0270	0.0542	0.0198	0.0366	0.0137	0.0279	0.0089	0.0166
Ran-S	0.0231	0.0520	0.0103	0.0207	0.0096	0.0210	0.0049	0.0099	0.0289	0.0563	0.0167	0.0300	**0.0184**	**0.0364**	0.0109	0.0208
CMR	0.0225	**0.0572**	0.0095	0.0192	0.0105	0.0209	0.0044	0.0094	0.0290	0.0562	0.0192	0.0374	0.0136	0.0278	0.0092	0.0167
CMRSI	0.0242	0.0555	0.0090	0.0183	0.0101	0.0214	0.0051	0.0104	0.0316	0.0603	0.0203	0.0395	0.0156	0.0310	0.0099	0.0173
BASRec	**0.0286**	0.0546	**0.0164**	**0.0307**	**0.0140**	**0.0289**	**0.0063**	**0.0128**	**0.0455**	**0.0810**	**0.0242**	**0.0436**	0.0164	0.0326	**0.0128**	**0.0223**

Table 3: Performance comparison of heuristic augmentation methods and BASRec on four datasets. All improvements are statistically significant, as determined by a paired t-test with the second best result in each case $(p \leq 0.05)$.

Evaluation Settings. We adopt the leave-one-out strategy to partition each user's item sequence into training, validation, and test sets. We rank the prediction over the whole item set rather than negative sampling, otherwise leading to biased discoveries (Krichene and Rendle 2020). The evaluation metrics include Hit Ratio@K (denoted by H@K), and Normalized Discounted Cumulative Gain@K (N@K). We report results with K $\in \{10, 20\}$.

Main Results with Various Backbone Models

The experimental results of the original SR models and adding our BASRec are presented in Table 2. We can observe that BASRec can significantly improve the performance of various types of SR models. Average performance gains on GRU4Rec, NextItNet, SASRec, and FMLPRec were 72.04%, 21.76%, 33.84%, and 68.50%, respectively. This result shows that the samples synthesized by our method can enhance the model's ability to learn user preferences further. Our approach achieves a win-win situation for both relevance and diversity, significantly improving the performance while ensuring plug-and-play generalization. Besides, the performance of the original model shows that SASRec outperforms the other models overall, demonstrating the power of the transformer in sequence modeling. With BASRec, the SASRec and FMLP-Rec are on par with each other. We believe that different models may have different underutilized performance potential. Our approach further exploits this potential through multiple mixup strategies.

Method	Beauty		Sports		Yelp		Home	
	N@10	H@10	N@10	H@10	N@10	H@10	N@10	H@10
Base	0.0338	0.0639	0.0174	0.0320	0.0136	0.0277	0.0078	0.0149
ASReP	0.0351	0.0664	0.0195	0.0353	0.0162	0.0319	0.0099	0.0184
DiffuASR	0.0372	0.0679	0.0202	0.0387	0.0150	0.0308	0.0105	0.0179
CL4SRec	0.0366	0.0686	0.0221	0.0412	**0.0176**	**0.0355**	0.0119	0.0212
BASRec	**0.0455**	**0.0810**	**0.0242**	**0.0436**	0.0164	0.0326	**0.0128**	**0.0223**

Table 4: Performance comparison of training-required methods and BASRec. The backbone network is SASRec.

Comparison with Data Augmentation Methods

We compare BASRec with different augmentation methods. For heuristic methods, we choose two representative models, SASRec and GRU4Rec, as the backbone network. For methods that require training, we chose the SASRec since it is available as a backbone network for all baselines.

Heuristic Methods. Table 3 shows that BASRec outperforms existing heuristic augmentation methods in most cases. Our method generates new training samples by mixing representations in the representation space, which can better preserve the original sequence semantics and discover more cross-user preferences than existing methods. Among the baseline methods, CMRSI and Ran-S usually perform better. This suggests that well-designed data augmentation operators or augmentation using interactions in the original sequence are effective. In some cases, existing methods lead to model performance degradation. We believe this may be

Method	Beauty		Sports		Yelp		Home	
	N@10	H@10	N@10	H@10	N@10	H@10	N@10	H@10
Base	0.0338	0.0639	0.0174	0.0320	0.0136	0.0277	0.0078	0.0149
w/o SA	0.0415	0.0734	0.0209	0.0374	0.0152	0.0295	0.0097	0.0184
w/o ALW	0.0439	0.0778	0.0229	0.0415	0.0146	0.0297	0.0113	0.0201
w/o CA	0.0397	0.0721	0.0193	0.0353	0.0145	0.0290	0.0098	0.0170
w/o NL	0.0422	0.0752	0.0227	0.0412	0.0152	0.0300	0.0111	0.0197
w/o Two	0.0416	0.0742	0.0231	0.0411	0.0151	0.0304	0.0117	0.0207
BASRec	**0.0455**	**0.0810**	**0.0242**	**0.0436**	**0.0164**	**0.0326**	**0.0128**	**0.0223**

Table 5: Performance of different variants of BASRec. The backbone network is SASRec.

related to the fact that these methods introduce much noise into the augmented data, which interferes with model learning. In summary, it is essential to balance relevance and diversity in data augmentation, and favoring one side too much can lead to performance degradation.

Training-required Methods. Table 4 shows the performance of BASRec compared to the training-required baselines. These methods usually contain auxiliary tasks or data generation modules that require training. BASRec achieves the best or second-best results without increasing model parameters. CL4SRec usually performs best among the baseline methods, indicating that contrastive learning can effectively mine preference information from sparse data.

Ablation Study

We conduct an ablation study to explore the effectiveness of various components in our method. We compare our BASRec with the following variants: 1) w/o SA: remove the Single Augmentation. 2) w/o ALW: remove the Adaptive Loss Weighting in Single Augmentation. 3) w/o Out: remove the Cross Augmentation. 4) w/o NL: remove the Nonlinear Mixup strategy in Cross Augmentation, i.e., perform general linear Mixup. 5) w/o Two: Use Eq. 16 to jointly train the model from scratch without a two-stage training strategy.

The results are shown in Table 5. The performances decrease significantly after removing either the Single Augmentation or the Cross Augmentation, suggesting that data augmented by both modules contributes to model training. When we replace Adaptive Loss Weighting with consistent weights, the model is unable to measure the difference between the enhanced data and the original data. Learning and distinguishing this difference can improve the model's performance and robustness. Nonlinear Mixup further extends the possibility of Cross Augmentation to generate augmented samples, and nonlinear combinations between different samples can help the model learn cross-preferences and fine-grained preferences. Besides, jointly training from scratch results in inferior performance compared to BASRec in four datasets, highlighting the significance of the two-stage training procedure. When two-stage training is used, the model can learn accurate representations in the first stage, while the second stage produces high-quality augmented representations by mixing these representations. If Mixup is performed at the beginning of training, inaccurate representations may interfere with each other, which is detrimental to model learning and convergence.

Method	Beauty	Sports	Yelp	Home	Average
CMRSI	0.5673	0.7273	0.6324	0.6884	0.6539
ASReP	0.9585	0.9476	0.9307	0.9752	0.9530
CLS4Rec	0.9612	0.9574	0.9580	0.9631	0.9599
BASRec-I	0.9083	0.8954	0.8626	0.9522	0.9046
BASRec-O	0.9239	0.9172	0.9003	0.9460	0.9219
BASRec	0.9152	0.9076	0.8790	0.9485	0.9126

Table 6: Cosine similarity between the generated samples and the original samples. The backbone network is SASRec. The 'BASRec-I' and 'BASRec-O' represent only use Single-sequence augmentation and Cross-sequence augmentation, respectively.

Data Similarity Analysis

We calculated the cosine similarity between the samples generated by different data augmentation methods and the original samples. Since discrete interaction data cannot be computed directly, we compute the similarity of the final output representation of the model for all samples. We report the average similarity value throughout the training process and present the result in Table 6.

The table shows that the heuristic augmentation method CMRSI generates samples with relatively low similarity (i.e., lack of relevance), resulting in the loss of important preference knowledge and sequence semantics contained in the original samples. For training-required augmentation methods, ASReP employs a bi-directional Transformer to generating similar sequence data directly. The contrastive learning objective in CL4SRec draws close the distance between the original and generated samples. These methods result in high similarity between the augmented and original samples (i.e., lack of diversity). Our method generates samples with suitable similarity through a well-designed augmentation and fusion strategy. The new samples are diverse enough but retain the important semantics of the original data, balancing the relevance and diversity.

Conclusion

This paper introduces BASRec, a Balanced Data Augmentation Plugin for Sequential Recommendation, which aims to balance the relevance and diversity of augmented data. Our approach consists of Single-sequence Augmentation and Cross-sequence Augmentation. The former balances the two properties by heuristic operations with elaborate fusion and reweighting strategy. The latter fuses sequence representations from different users to generate samples that are diverse but retain important semantics of the original sequence. Extensive experiments demonstrate the superiority of BASRec. It improves the performance of various sequential models and outperforms existing methods. Our work emphasizes the importance of balancing relevance and diversity and demonstrates the great potential of augmentation in the representation space. For future work, we will further improve the method so that it can be integrated into other recommendation models. Moreover, we are interested in improving the simplicity of the method, e.g., by adaptively choosing operator rates and mixup weights.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under Grant No. 62032013, No. 62102074, and the Science and technology projects in Liaoning Province (No. 2023JH3/10200005).

References

Bian, S.; Zhao, W. X.; Wang, J.; and Wen, J.-R. 2022. A relevant and diverse retrieval-enhanced data augmentation framework for sequential recommendation. In *CIKM*, 2923–2932.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, 1597–1607. PMLR.

Cheng, M.; Zhang, H.; Liu, Q.; Yuan, F.; Li, Z.; Huang, Z.; Chen, E.; Zhou, J.; and Li, L. 2024. Empowering Sequential Recommendation from Collaborative Signals and Semantic Relatedness. *arXiv preprint arXiv:2403.07623*.

Dang, Y.; Liu, Y.; Yang, E.; Guo, G.; Jiang, L.; Wang, X.; and Zhao, J. 2024a. Repeated Padding as Data Augmentation for Sequential Recommendation. *arXiv preprint arXiv:2403.06372*.

Dang, Y.; Yang, E.; Guo, G.; Jiang, L.; Wang, X.; Xu, X.; Sun, Q.; and Liu, H. 2023a. TiCoSeRec: Augmenting Data to Uniform Sequences by Time Intervals for Effective Recommendation. *TKDE*.

Dang, Y.; Yang, E.; Guo, G.; Jiang, L.; Wang, X.; Xu, X.; Sun, Q.; and Liu, H. 2023b. Uniform sequence better: Time interval aware data augmentation for sequential recommendation. In *AAAI*, volume 37, 4225–4232.

Dang, Y.; Yang, E.; Liu, Y.; Guo, G.; Jiang, L.; Zhao, J.; and Wang, X. 2024b. Data Augmentation for Sequential Recommendation: A Survey. *arXiv preprint arXiv:2409.13545*.

Fan, Z.; Liu, Z.; Wang, Y.; Wang, A.; Nazari, Z.; Zheng, L.; Peng, H.; and Yu, P. S. 2022. Sequential recommendation via stochastic self-attention. In *WWW*, 2036–2047.

Guo, H. 2020. Nonlinear mixup: Out-of-manifold data augmentation for text classification. In *AAAI*, volume 34, 4044–4051.

He, R.; and McAuley, J. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *ICMD*, 191–200. IEEE.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Hu, H.; He, X.; Gao, J.; and Zhang, Z.-L. 2020. Modeling personalized item frequency information for next-basket recommendation. In *SIGIR*, 1071–1080.

Huang, T.; Dong, Y.; Ding, M.; Yang, Z.; Feng, W.; Wang, X.; and Tang, J. 2021. Mixgcf: An improved training method for graph neural network-based recommender systems. In *KDD*, 665–674.

Jiang, J.; Luo, Y.; Kim, J. B.; Zhang, K.; and Kim, S. 2021. Sequential recommendation with bidirectional chronological augmentation of transformer. *arXiv preprint arXiv:2112.06460*.

Jing, M.; Zhu, Y.; Zang, T.; and Wang, K. 2023. Contrastive self-supervised learning in recommender systems: A survey. *TOIS*, 42(2): 1–39.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *ICMD*, 197–206. IEEE.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krichene, W.; and Rendle, S. 2020. On sampled metrics for item recommendation. In *KDD*, 1748–1757.

Liu, Q.; Wu, S.; Wang, D.; Li, Z.; and Wang, L. 2016. Context-aware sequential recommendation. In *ICDM*, 1053–1058. IEEE.

Liu, Q.; Yan, F.; Zhao, X.; Du, Z.; Guo, H.; Tang, R.; and Tian, F. 2023. Diffusion augmentation for sequential recommendation. In *CIKM*, 1576–1586.

Liu, Z.; Chen, Y.; Li, J.; Yu, P. S.; McAuley, J.; and Xiong, C. 2021. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479*.

Luo, T.; Liu, Y.; and Pan, S. J. 2024. Collaborative Sequential Recommendations via Multi-View GNN-Transformers. *TOIS*.

McAuley, J.; Pandey, R.; and Leskovec, J. 2015. Inferring networks of substitutable and complementary products. In *KDD*, 785–794.

Qiu, R.; Huang, Z.; Yin, H.; and Wang, Z. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *WSDM*, 813–823.

Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, 811–820.

Su, J.; Chen, C.; Lin, Z.; Li, X.; Liu, W.; and Zheng, X. 2023. Personalized Behavior-Aware Transformer for Multi-Behavior Sequential Recommendation. In *MM*, 6321–6331.

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*, 1441–1450.

Tan, Y. K.; Xu, X.; and Liu, Y. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st workshop on deep learning for recommender systems*, 17–22.

Tang, J.; and Wang, K. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*, 565–573.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *NIPS*, 30.

Wang, J.; Le, Y.; Chang, B.; Wang, Y.; Chi, E. H.; and Chen, M. 2022. Learning to augment for casual user recommendation. In *WWW*, 2183–2194.

Wang, Z.; Zhang, J.; Xu, H.; Chen, X.; Zhang, Y.; Zhao, W. X.; and Wen, J.-R. 2021. Counterfactual data-augmented sequential recommendation. In *SIGIR*, 347–356.

Xie, X.; Sun, F.; Liu, Z.; Wu, S.; Gao, J.; Zhang, J.; Ding, B.; and Cui, B. 2022. Contrastive learning for sequential recommendation. In *ICDE*, 1259–1273. IEEE.

Yang, H.; Choi, Y.; Kim, G.; and Lee, J.-H. 2023a. LOAM: improving long-tail session-based recommendation via niche walk augmentation and tail session mixup. In *SIGIR*, 527–536.

Yang, Y.; Huang, C.; Xia, L.; Huang, C.; Luo, D.; and Lin, K. 2023b. Debiased contrastive learning for sequential recommendation. In *WWW*, 1063–1073.

Yuan, F.; Karatzoglou, A.; Arapakis, I.; Jose, J. M.; and He, X. 2019. A simple convolutional generative network for next item recommendation. In *WSDM*, 582–590.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *ICLR*.

Zhang, L.; Zhou, X.; and Shen, Z. 2023. Multimodal pretraining framework for sequential recommendation via contrastive learning. *arXiv preprint arXiv:2303.11879*.

Zhang, R.; Yu, Y.; and Zhang, C. 2020. SeqMix: Augmenting Active Sequence Labeling via Sequence Mixup. In *EMNLP*, 8566–8579.

Zhao, C.; Li, X.; He, M.; Zhao, H.; and Fan, J. 2023. Sequential Recommendation via an Adaptive Cross-domain Knowledge Decomposition. In *CIKM*, 3453–3463.

Zhou, K.; Yu, H.; Zhao, W. X.; and Wen, J.-R. 2022. Filter-enhanced MLP is all you need for sequential recommendation. In *WWW*, 2388–2399.

Appendix

Details for Implementations and Baselines

All experiments are performed on a single NVIDIA RTX 3090Ti GPU with an Intel Core i7-12700 CPU and 32GB RAM. All models are implemented using the PyTorch framework version 1.12.1+cu116.

The baselines consist of three categories. The first category is general models to validate the effectiveness of BASRec. These models employ diverse architectures, including RNN, CNN, Transformer, and MLP. The second category is heuristic augmentation methods. The third category is training-required augmentation models.

General Models

- **GRU4Rec** (Hidasi et al. 2015): This model leverages gated recurrent units to capture behavioral patterns.
- **NextItNet** (Yuan et al. 2019): This model combines masked filters with 1D dilated convolutions to model the long-range dependencies.
- **SASRec** (Kang and McAuley 2018): It adopts the multi-head self-attention mechanism to perform sequential recommendation.
- **FMLPRec** (Zhou et al. 2022): It is an all-MLP model with learnable filters for sequential recommendation.

Heuristic Methods

- **Random (Ran)** (Liu et al. 2023): This method augments each sequence by randomly selecting items from the whole item set
- **Random-seq (Ran-S)** (Liu et al. 2023): It selects items from the original sequence randomly as the augmentation items.
- **Slide Windows (SW)** (Tang and Wang 2018): It adopts slide windows to intercept multiple new subsequences from the original sequence.
- **CMR** (Xie et al. 2022): This work proposes three sequence data augmentation operators with contrastive learning. In this category, we only use the three operators, including Crop, Mask, and Reorder.
- **CMRSI** (Liu et al. 2021): Based on CMR, this work proposes two informative augmentation operators with contrastive learning. We adopt five operators including Crop, Mask, Reorder, Substitute, and Insert.

Training-required Methods

- **ASReP** (Jiang et al. 2021): This method employs a reversely pre-trained transformer to generate pseudo-prior items for short sequences. Then, fine-tune the pre-trained transformer to predict the next item.
- **DiffuASR** (Liu et al. 2023): It adopts the diffusion model for sequence generation. Besides, two guide strategies are designed to control the model to generate the items corresponding to the raw data.
- **CL4SRec** (Xie et al. 2022): This method leverages random data augmentation and utilizes contrastive learning to extract self-supervised signals from the original and augmented data.

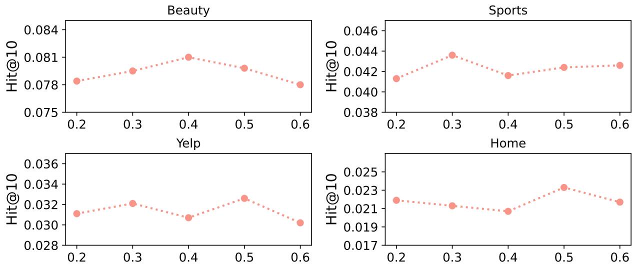

Figure 3: Sensitivity analysis of parameter α.

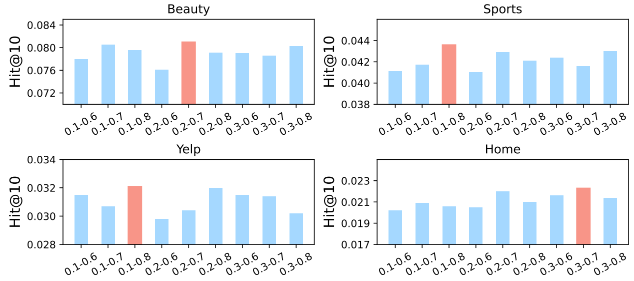

Figure 4: Sensitivity analysis of parameters a and b. For example, '$0.1 - 0.6$' represents $a = 0.1$ and $b = 0.6$. We highlight the best result with a different color.

We do not include methods ReDA (Bian et al. 2022), CASR (Wang et al. 2021), and L2Aug (Wang et al. 2022) since they do not provide open-source codes for reliable reproduction (empty code repository or no available code repository links).

Hyper-parameter Analysis

We further investigate the hyper-parameter α of beta distribution and a, b of the operators. The results are illustrated in Figure 3 and 4. In our approach, all beta distributions share the same α, and the two operators share the same a and b.

For α, we find that optimal values tend to arise in $\{0.3, 0.4, 0.5\}$, with both larger and smaller values leading to degradation of model performance. The optimal α for the four datasets is $0.4, 0.3, 0.5, 0.5$, respectively. For the a and b of operators, we observe that the optimal sequence operation ranges are concentrated in the middle. In other words, neither too many nor too few changes can be made to the original sequence. Also, a narrower range, such as only 0.4 for the '$0.2 - 0.6$', can lead to poorer performance. Overall, the choice of a and b needs to avoid values that are too small and too large but also allow sufficient room for choice between a and b. This finding corroborates the need to balance relevance and diversity during data augmentation.