Sequential Recommendation with Bidirectional Chronological Augmentation of Transformer

Juyong Jiang[1*], **Yingtao Luo**[2*], **Jae Boum Kim**[4], **Kai Zhang**[3], **Sunghun Kim**[4†]

[1]Upstage [2]University of Washington [3]Temple University
[4]Hong Kong University of Science and Technology
{csjuyongjiang, kzhang980}@gmail.com, yl3851@uw.edu, {jbkim, hunkim}@cse.ust.hk

Abstract

Sequential recommendation can capture user chronological preferences from their historical behaviors, yet the learning of short sequences is still an open challenge. Recently, data augmentation with pseudo-prior items generated by transformers has drawn considerable attention in improving recommendation in short sequences and addressing the cold-start problem. These methods typically generate pseudo-prior items sequentially in reverse chronological order (i.e., from the future to the past) to obtain longer sequences for subsequent learning. However, the performance can still degrade for very short sequences than for longer ones. In fact, reverse sequential augmentation does not explicitly take into account the forward direction, and so the underlying temporal correlations may not be fully preserved in terms of conditional probabilities. In this paper, we propose a Bidirectional Chronological Augmentation of Transformer (BiCAT) that uses a forward learning constraint in the reverse generative process to capture contextual information more effectively. The forward constraint serves as a bridge between reverse data augmentation and forward recommendation. It can also be used as pretraining to facilitate subsequent learning. Extensive experiments on two public datasets with detailed comparisons to multiple baseline models demonstrate the effectiveness of our method, especially for very short sequences (3 or fewer items).

Introduction

Recommender systems are a key engine underlying B2C commercial platforms such as the YouTube video and the Amazon shopping recommendation. Sequential recommendation, in particular, represents recommender systems that learn from historical user behaviors to capture the sequential preference and to infer the next possible item (Liu et al. 2016; Yu et al. 2016). A fundamental feature of sequential recommendation is the sequential order of user actions, which characterizes the influence of past behaviors on the current and future one (Donkers, Loepp, and Ziegler 2017; Ma, Kang, and Liu 2019). Based on this, transformer models such as SASRec (Kang and McAuley 2018) and BERT4Rec (Sun et al. 2019) have shown extraordinary performance in many sequential recommendation tasks.
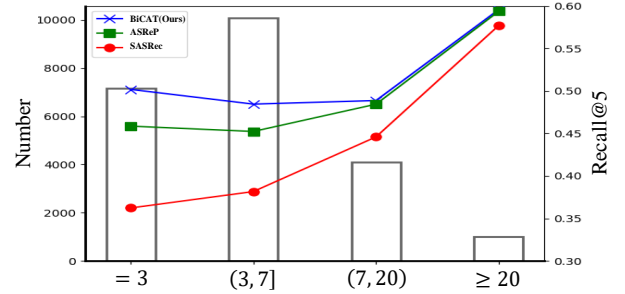
Figure 1: The Recall@5 performance w.r.t sequence length distribution (bar) on `Amazon Beauty` when using (1) no sequence augmentation (SASRec, red dots); (2) reverse sequence augmentation (ASReP, green squares) and (3) our forward-constrained reverse augmentation (BiCAT, blue stars, being particularly advantageous on short sequences).

Despite the success of state-of-the-art transformers in the sequential recommendation, researchers have reported that they are not effective on short sequences (Liu et al. 2021b). With limited information about the user sequential patterns, short sequences are much harder to predict, which is also known as the *cold-start* issue (Li et al. 2019a; Wang, Ding, and Caverlee 2021). Yet in the meantime, short sequences can be dominant in real-world datasets. As shown in the exemplified histogram of the Amazon Beauty dataset in Figure 1, 75% sequences have less than 7 items and only 5% sequences have more than 20 items.

To address this problem, data augmentation methods (Li et al. 2017; Yuan et al. 2020; Wang et al. 2021) are typically adopted to increase the length of the training sequences. Early augmentations methods (Xie et al. 2020; Zhou et al. 2020) such as crop, mask and reorder may break the sequential property. Therefore, it could exaggerate the cold-start issues since short sequences are vulnerable to the slight change of a few items (Liu et al. 2021a).

Recently, ASReP (Liu et al. 2021b) was proposed to learn the reverse sequential correlation (i.e., future-to-past) from data to generate pseudo-prior items at the beginning of the original sequences. Given the length-augmented sequences, it then fine-tunes the transformer from the original direction (i.e., past-to-future) to predict the next items. We illustrate the Recall@5 for the next-item prediction on Ama-
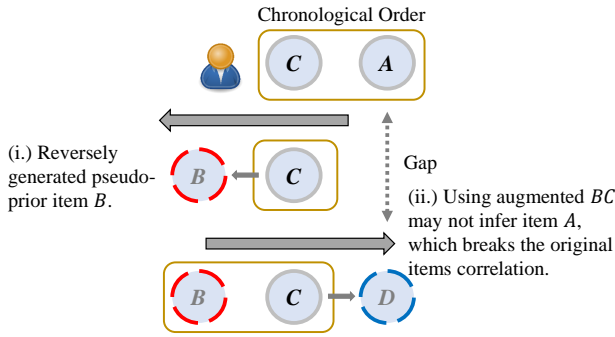
Figure 2: The optimization gap between the correlations of reversely augmented sequence and original sequence.

zon Beauty dataset with and without ASReP in Figure 1. As noted, recommendation performance on short sequences is worse than on long sequences. Although ASReP can improve the learning of short sequences, the performance on short sequences is still worse than on long sequences.

ASReP has an advantage over other methods in that it does not change the original sequences and only fills in the prior "missing" items. However, the augmentation process is a backward process (future to past), so the temporal correlations underlying the generated sequence may not be fully consistent with the original sequence in terms of the forward direction (past to future). As shown in Figure 2, although the reverse learning can maximize the reverse sequential probability, the generated item B may not fit in the original sequential correlation to maximize the forward probability to discover the future ground-truth item A. To further illustrate the reasoning, we provide theoretical explanations in the methodology section with a counterexample in Figure 3.

Since the reverse augmentation of ASReP may not fit in the correlation of the forward recommendation step, to preserve the sequential correlation in the augmentation process, we propose a Bidirectional Chronological Augmentation of Transformer (BiCAT) that uses a forward learning constraint in the loss to capture the contextual information when generating items reversely. Namely, when using the newly generated items to predict their subsequent items, such an extra representation learning condition in the augmentation phase is more consistent with the forward recommendation phase, which we believe could lead to better generalization performance. In the meantime, BiCAT can also serve as a better pretraining for the recommendation task.

We verify the effectiveness of our BiCAT by reproducible experimental results on two public datasets. Comparisons to multiple baseline models show that BiCAT can improve the learning of short sequences by a large margin and accelerate the training. The main contributions of this paper are:

- We propose Bidirectional Chronological Augmentation of Transformer (BiCAT) that adds a novel forward learning constraint in the loss to preserve the original correlation when generating items reversely, which bridges the gap between augmentation and recommendation.

- We prove theoretically that the sequential correlations of two different chronological directions may not necessar-

ily be equivalent, which explains why reverse data augmentation alone may produce sub-optimal results.

- We give extensive experiments to show that BiCAT can improve the transformer on public datasets significantly. For short sequences with lengths smaller than 3 in particular, BiCAT can improve the transformer by 40.12% on average and improve the ASReP augmented model by 6.71% on average. A fine-tuning comparison shows that BiCAT pre-training converges faster than ASReP.

Related Work

Sequential recommendation

The core of sequential recommendation is the modeling of user-item interaction pattern and the prediction of future items based on each user's historical behaviors. Early works (Koren, Bell, and Volinsky 2009; Rendle, Freudenthaler, and Schmidt-Thieme 2010; He and McAuley 2016) mainly reply on Markov chains to model the pair-wise item transition. Later, with the rise of deep learning, Markov models are extensively replaced by neural networks. Recurrent neural networks (RNNs) (Hidasi et al. 2015; Yu et al. 2016; Liu et al. 2016) that can store the sequential memory are widely used in sequential recommendations. RNNs with invariant structures and/or gating mechanisms (Quadrana et al. 2017; Ma, Kang, and Liu 2019; Song et al. 2019) are also proposed for different recommendation tasks. Other improvements for RNNs are also studied, such as using contextual features (Smirnova and Vasile 2017) and alternative sampling tricks (Hidasi and Karatzoglou 2018). Convolutional neural networks (CNNs) (Tang and Wang 2018; Yuan et al. 2019) are also proposed to focus on learning the local transitions. These models with limited reception fields still faces the long-term dependency problem. More recently, attention-based models (Kang and McAuley 2018; Kang et al. 2019a; Sun et al. 2019; Ma et al. 2020b; Luo, Liu, and Liu 2021) are adopted in sequential recommendation tasks, which achieve outstanding performance. For example, SASRec applies the transformer layer from NLP tasks to adapt to sequential recommendation. BERT4Rec further considers a bidirectional transformer layer. Additionally, other techniques such as relation awareness (Ji et al. 2020), graph learning (Ma et al. 2020a) and reinforcement learning (Xin et al. 2020) are also used to adapt to various sequential recommendation tasks.

Data augmentation

Data augmentation is a useful component in machine learning venues to increase the volume of data and hence the model performance. In the literature of recommender systems, many kinds of data augmentation methods are proposed. For example, (Tan, Xu, and Liu 2016) uses data augmentation of sequence preprocessing and embedding dropout to enhance training and reduce overfitting. (Yuan et al. 2020) proposes to integrate future interaction as a data augmentation to improve model training. Generative augmentations (Wang et al. 2019; Li et al. 2019b) are proposed to add extra user-item interaction data by generative adversarial networks or Seq2Seq models. A rising technique that can be classified as data augmentation is self-supervised

learning for sequential recommendation (Zhou et al. 2020; Yao et al. 2020; Xie et al. 2020; Liu et al. 2021a), which uses operations such as crop, reorder, mask, substitute, and insert to supplement the original sequences. These methods are designed for improving model robustness, but may not address the cold-start problems since they may inevitably break the sequential property of the original sequences and the learning of short sequences is vulnerable to the change of a few items (Liu et al. 2021a). Recently, a reverse data augmentation method ASReP is proposed for the cold-start learning of short sequences, which adds pseudo-prior items in the beginning of the original sequences by learning to predict the past items reversely. Traditionally, cold-start problems are mainly addressed by using side information (Saveski and Mantrach 2014), knowledge transfer (Kang et al. 2019b), and meta-learning (Dong et al. 2020; Zheng et al. 2021).

Methodology

Chronological correlation

In this section, we prove by Bayes' theorem that the forward sequential correlation is not necessarily consistent with the reverse sequential correlation, so the reverse augmentation may lead to sub-optimal augmentation. Assume that A and B are two items interacting with a user chronologically. Since the user interacts with A first, we denote the forward correlation as $P(B|A)$ and write $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$ where $P(B) \neq 0$. Thus, the reverse correlation $P(A|B)$ equals the forward correlation $P(B|A)$ only when $P(A) = P(B)$. It indicates that the two items always appear concurrently, which is not always true in practical item sequences.

We illustrate this with an example in Figure 3. We define three events (i.e., A, B, C), and two probabilities (i.e., forward probability P_F and reverse probability P_R). We consider four sequences as the training data in Figure 3 (a) to indicate the various inference probabilities of the model. For instance, to calculate the forward correlation $P_F(A|C)$, we count the cases with A on the right side and C on the left side (i.e., "CA" in the forward order) and divide it by the number of times with A on the right (i.e., "XA" where X can be any events). In this case, $P_F(A|C) = P_F(C, A)/P_F(C) = 3/4$ with three "CA" and one "CD" as shown in Figure 3 (b)(i.).

Now, if we use C to predict the next item A in the forward recommendation task, what would be the best choice for augmenting C with a prior (left) item? By reverse inference, $P_R(B|C) = 3/4$ is the largest among other possibilities, so the reverse augmentation model should choose B as shown in Figure 3 (b)(ii.). However, does B improve the model performance? By forward inference on top of an augmented "BC", we find that $P_F(A|CB) = 2/3 < 3/4$, which is, in fact, worse than the original performance. This counterexample shows that reverse augmentation does not always improve the learning of the original sequential correlation.

In light of this challenge, we can also consider a different constraint that chooses the left element by maximizing the forward correlation instead of the reverse correlation. As shown in Figure 3 (b)(iii.), we find that D can achieve that and make the forward recommendation more accurate with $P_F(A|CD) = 1/1 > 3/4$. This example shows that the
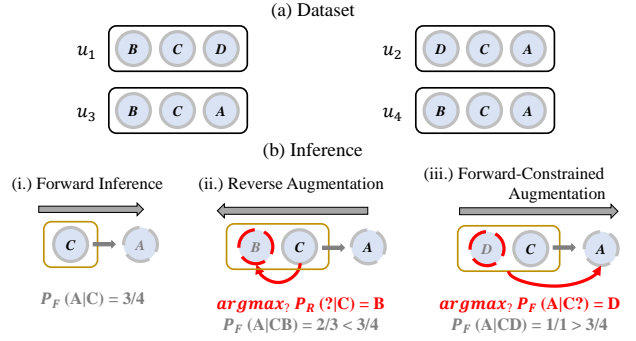

(a) Dataset

(b) Inference

Figure 3: A counterexample that the reverse augmentation could break the original forward sequential correlation and result in the decrease of model predictive performance.

forward-constrained augmentation can effectively preserve of the original forward sequential correlation. We formally introduce our method in the following sections.

Problem Statement

Nnotations. $\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$ denotes a set of users, $\mathcal{V} = \{v_1, v_2, ..., v_{|\mathcal{V}|}\}$ denotes a set of items, and $\mathcal{S}_u = [v_1^{(u)}, v_2^{(u)}, ..., v_n^{(u)}]$ denotes the interaction sequence in chronological order of user $u \in \mathcal{U}$. Each $v_i^{(u)}$ refers to an item in $\mathcal{V}$ that u has interacted with at time step i. Sequential recommendation aims to predict the item that user u will interact with at the $(n+1)$th step. The probability of each item $v \in \mathcal{V}$ becoming the $(n+1)$th item is denoted as $p(v_{n+1}^{(u)} = v|S_u)$, given the training sequence S_u.

Model Architecture

Here, we introduce an effective data augmentation model of pseudo-prior items for short sequences, termed **B**idirectional **C**hronological **A**ugmentation of **T**ransformer (**BiCAT**). It is built upon the prior work (Liu et al. 2021b) and (Kang and McAuley 2018). The proposed BiCAT consist of 1) an embedding layer; 2) multi-stacked unidirectional transformer layers; and 3) a bidirectional chronological pre-trained transformer for short sequences augmentation. With these modules, the left-to-right transformer is fine-tuned using the augmented sequences for the next item prediction. Our method employs a forward learning constraint in data augmentation to capture the contextual information when generating items reversely. The model architecture is shown in Figure 4.

Embedding Layer

In standard Transformer (Vaswani et al. 2017), the embedding layer consists of item embedding and positional embedding. The item embedding layer encodes each item ID $v_i \in \mathbb{R}^1$ into a latent representation $e_i \in \mathbb{R}^d$. The item embedding transforms the scalar item IDs into dense vectors, aiming to reduce computation and improve representation learning. To make a fair comparison, we follow (Liu et al. 2021b) to hold
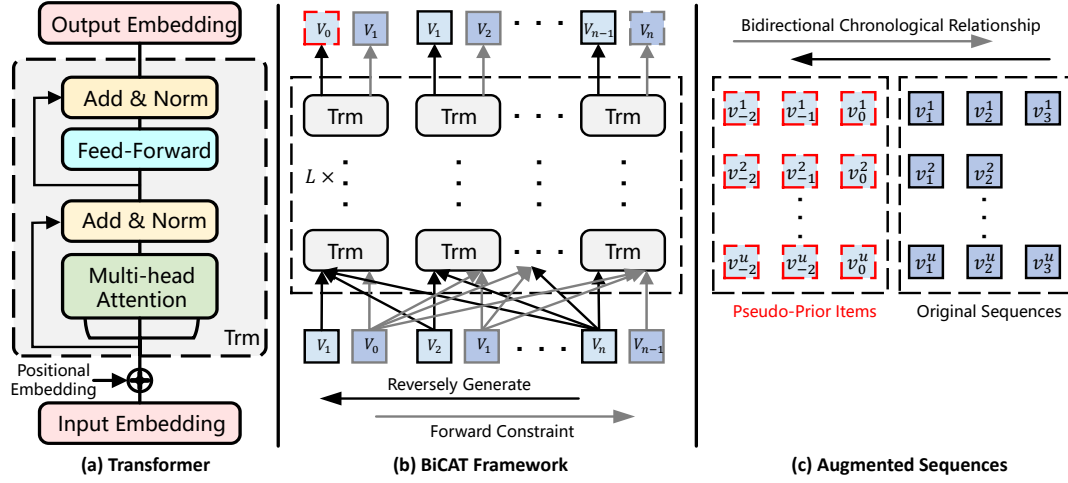
Figure 4: The model architecture of the proposed BiCAT.

a position embedding table $P \in \mathbb{R}^{d \times n}$, where n is the maximum sequence length and d is the same embedding dimensions of the item embedding. Note that for a sequence S_u, we truncate it to be the last n items if its length is longer than n or utilize zero-padding if its length is shorter than n (Kang and McAuley 2018). Overall, the embedding layer transforms the sequence of $\mathcal{S}_u = [v_1^{(u)}, v_2^{(u)}, ..., v_n^{(u)}] \in \mathbb{R}^n$ into $E(S_u) = [e_1^{(u)} + p_1^{(u)}, e_2^{(u)} + p_2^{(u)}, ..., e_n^{(u)} + p_n^{(u)}] \in \mathbb{R}^{n \times d}$.

Transformer Layer

As shown in Figure 4 (a), the basic Transformer layer Trm is composed of two sub-layers, a *Multi-Head Self-Attention* sub-layer and a *Position-wise Feed-Forward Network*.

Multi-Head Self-Attention Specifically, the Multi-Head Self-Attention sub-layer adopts a multi-head attention layer and then a normalization operation on the embedded sequence. Each attention layer adopts the scaled dot-product attention layer used in (Vaswani et al. 2017; Sun et al. 2019; Wu et al. 2020; Kang and McAuley 2018; Liu et al. 2021b), which potentially learn the different importance pattern of items in sequences from h sub-spaces.

$$\text{H} = \text{MH}(E) = \text{Concat}[\text{head}_1, ..., \text{head}_h]W^O, \quad (1)$$

$$\text{head}_i = \text{Att}(EW_i^Q, EW_i^K, EW_i^V), \quad (2)$$

where

$$\text{Att}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}})V. \quad (3)$$

Here, $E = E(S_u) \in \mathbb{R}^{n \times d}$ is the embedded interaction sequence. Note that we omit the layer subscript l for convenience. h denotes the number of attention layers. $W_i^Q \in \mathbb{R}^{d \times d/h}$, $W_i^K \in \mathbb{R}^{d \times d/h}$, $W_i^V \in \mathbb{R}^{d \times d/h}$, $W_i^O \in \mathbb{R}^{d \times d}$ denote four different and learnable parameters that process the *query* Q, *key* K, *value* V, and the output of the multi-head attention layer. The effect of temperature $\sqrt{d}$ is to avoid

overly large values of the inner product and extremely small gradients (Hinton, Vinyals, and Dean 2015; Vaswani et al. 2017; Kang and McAuley 2018; Sun et al. 2019). The tensor dimension of the final output of the multi-head attention sub-layer is still maintained as $\mathbb{R}^{n \times d}$.

Position-wise Feed-Forward Network To enhance the representation learning capability, we use a *Position-wise Feed-Forward Network* to project the output of the multi-head attention sub-layer to the sequence embedding at each position i. Specifically, the FFN contains two affine transformations layers with a ReLU activation function, as used in all the prior works (Vaswani et al. 2017; Kang and McAuley 2018; Liu et al. 2021b).

$$\text{PFFN}(H) = \text{Concat}[\text{FFN}(h_i)^T, ..., \text{FFN}(h_n)^T]^T, \quad (4)$$

$$\text{FFN}(x) = \text{ReLU}(xW^{(1)} + b^{(1)})W^{(2)} + b^{(2)}, \quad (5)$$

where $W^{(1)} \in \mathbb{R}^{d \times d}$, $W^{(2)} \in \mathbb{R}^{d \times d}$, $b^{(1)} \in \mathbb{R}^d$, and $b^{(2)} \in \mathbb{R}^d$ are learnable parameters and shared across all positions. We omit the layer subscript l for the sake of simplicity.

Bidirectional Chronological Pre-training

We use the transformer in SASRec (Kang and McAuley 2018) as our backbone, as shown in the left-hand side of Figure 4. First, we reverse the original input sequence of $\mathcal{S}_u = [v_1^{(u)}, v_2^{(u)}, ..., v_n^{(u)}] \in \mathbb{R}^n$ into $\overleftarrow{\mathcal{S}}_u = [v_n^{(u)}, v_{n-1}^{(u)}, ..., v_1^{(u)}] \in \mathbb{R}^n$. Then, we inject the $\overleftarrow{\mathcal{S}}_u$ into the Transformer to generate the next items as the pseudo-prior items for short sequences. This reversely generation (inference) can be formulated by:

$$p(v_0^{(u)} = v_0 | \overleftarrow{\mathcal{S}}_u), \quad (6)$$

where $v_0 \in \mathcal{V}$ denotes the pseudo-prior items. In the training phase, we mask the left items in $\overleftarrow{\mathcal{S}}_u$ by zero-padding, use them as labels, and train the Transformer to predict them.

The reverse learning of the Transformer can capture the reverse item correlations (Sun et al. 2019; Liu et al. 2021b) and learn a reversely chronological user's preferences.

However, as shown in Figures 2 and 3, the reverse augmentation may be inconsistent with the original correlation. Herein, we propose the forward learning constraint, which bridges the gap between reverse augmentation and forward recommendation. Specifically, the forward constraint leverages the generated pseudo-prior item v_0 to predict v_n item in the forward direction in parallel with Eq. (6). Formally, we feed the augmented sequence $\overrightarrow{\mathcal{S}}_u = [v_0^{(u)}, v_1^{(u)}, ..., v_{n-1}^{(u)}] \in \mathbb{R}^n$ into the identical Transformer to concurrently maximize the future item prediction

$$p(v_n^{(u)} = v_n | \overrightarrow{S}_u), \qquad (7)$$

where $v_n \in \mathcal{V}$ denotes the future items of the original sequence at position n. Similarly, we also mask the right items at position n in $\overrightarrow{S}_u$ by zeros and train the Transformer to predict them. Since we unearth Bi-Chronological user's preferences from their historical behaviors simultaneously, we named our augmentation model **BiCAT** in short.

Prediction Layer After L layers adaptively and hierarchically aggregate information across all positions, we get the final output $\mathbf{H}^L \in \mathbb{R}^{n \times d}$ for all items of the input sequence. Then we apply an matrix factorization (MF) (Koren, Bell, and Volinsky 2009) layer to predict the relevance of item v_i:

$$r_{i,j} = \mathbf{H}_j^L \mathbf{N}_i^{\text{table}}, \qquad (8)$$

where $r_{i,j}$ is the relevance of item v_i being the next item given the first j items (i.e., $v_1, v_2, ..., v_j$), and $\mathbf{N}^{\text{table}} \in \mathbb{R}^{d \times |\mathcal{V}|}$ denotes an item embedding table. Note that we use shared item embedding table in the input and output layer.

Network Training With the bidirectional chronological losses, the model has a stronger learning criterion to achieve a better performance. In this work, following SASRec (Kang and McAuley 2018), we adopt the binary cross entropy loss combined by losses of both *left* and *right* directions:

$$\mathcal{L}_{total} = \overleftarrow{L}_l + \lambda \overrightarrow{L}_r, \qquad (9)$$

where

$$\overleftarrow{L}_l = -\sum_{i=1}^{\mathcal{U}} \sum_{j=n}^{1} \left[log(\sigma(r_{o_j,j})) + \sum_{t \notin \overleftarrow{S}_i} log(1 - \sigma(r_{t,j})) \right],$$

$$\overrightarrow{L}_r = -\sum_{i=1}^{\mathcal{U}} \sum_{j=0}^{n-1} \left[log(\sigma(r_{o_j,j}) + \sum_{t \notin \overrightarrow{S}_i} log(1 - \sigma(r_{t,j})) \right].$$

Here, λ is a hyper-parameter to balance the forward constraint and the reverse generation. It is set as 1 by default. Our model takes $\overleftarrow{\mathcal{S}}_u = [v_n^{(u)}, v_{n-1}^{(u)}, ..., v_1^{(u)}] \in \mathbb{R}^n$ and $\overrightarrow{\mathcal{S}}_u = [v_0^{(u)}, v_1^{(u)}, ..., v_{n-1}^{(u)}] \in \mathbb{R}^n$ as inputs. We denote the corresponding sequences $\overleftarrow{\mathcal{O}}_u = [o_{n-1}^{(u)}, o_{n-2}^{(u)}, ..., o_0^{(u)}] \in \mathbb{R}^n$ and $\overrightarrow{\mathcal{O}}_u = [o_1^{(u)}, o_2^{(u)}, ..., o_n^{(u)}] \in \mathbb{R}^n$ as the outputs.

Short Sequence Augmentation After the training of the bidirectional chronological pre-trained Transformer, following (Liu et al. 2021b), we use it to recursively generate pseudo-prior items, as illustrated in Figure 4 (b). To be specific, we use K to denote the number of augmented items and M to denote the length threshold of whether a sequence

---

Algorithm 1: Recursive Generation.

Input : The number of augmented items K, the length threshold M, the pre-trained Transformer BiCAT, and the reverse sequence $\overleftarrow{\mathcal{S}}_u = [v_n^{(u)}, v_{n-1}^{(u)}, ..., v_1^{(u)}]$.

Output: $[v_{-K+1}^{(u)}, \ldots, v_{-1}^{(u)}, v_0^{(u)}]$.

for $k \in \{1, 2, ..., K\}$ **do**
  if $|\overleftarrow{S}_u| < M$ **then**
    $v_{-(k-1)}^{(u)} = \text{BiCAT}(\overleftarrow{\mathcal{S}}_u = [v_n^{(u)}, v_{n-1}^{(u)}, ..., v_1^{(u)}, v_0^{(u)}, v_{-1}^{(u)}, ..., v_{-(k-2)}^{(u)}]);$
    $\overleftarrow{S}_u = \overleftarrow{S}_u \cup v_{-(k-1)}^{(u)};$
    add $v_{-(k-1)}^{(u)}$ to $[v_{-(k-2)}^{(u)}, \ldots, v_{-1}^{(u)}, v_0^{(u)}];$
  else
   | Continue
  end
end

---

is a "short sequence". Formally, we indicate the K augmented pseudo-prior items as $[v_{-K+1}^{(u)}, \ldots, v_{-1}^{(u)}, v_0^{(u)}]$ and arrange them at the beginning of the original short sequence $\mathcal{S}_u = [v_1^{(u)}, v_2^{(u)}, ..., v_n^{(u)}] \in \mathbb{R}^n$ for the next fine-tuning stage, as shown in Figure 4 (c). The recursive generation of pre-trained BiCAT is presented in Algorithm 1.

Left-to-Right Fine-tuning

Finally, we fine-tune the pre-trained transformer with the augmented sequences to predict the next item v_{n+1} in the left-to-right direction. Note that the forward learning constraint is only used in the pseudo-prior item generation stage, not in the fine-tuning stage. The fine-tuning process of our BiCAT is identical to that of ASReP (Liu et al. 2021b), except that our pre-trained transformer learns forward sequential pattern that is in line with the recommendation task.

Experiments

In this section, we present our experimental setup and evaluate the BiCAT on two public benchmarks. We answer the following Research Questions (RQs):

- **RQ1:** Does BiCAT outperform the state-of-the-art sequential recommendation models?

- **RQ2:** What is the influence of key hyperparameters in the BiCAT architecture?

- **RQ3:** How does the BiCAT perform in particular on short sequences?

Table 1: Statistics of datasets.

Datasets	#reviews	#users	#items	density
Beauty	198,502	22,363	12,101	0.07%
Phones	194,439	27,879	10,429	0.06%

- **RQ4:** Does BiCAT provide better pre-train model for the next left-to-right fine-tuning stage?

Datasets

In this paper, our main contribution is the more effective data augmentation. To compare fairly with prior works to show the effectiveness of our method, we use the same datasets as used in (Liu et al. 2021b). Amazon[1] is a series of datasets introduced in (Kang and McAuley 2018) which consist of large corpora of product reviews crawled from Amazon.com by McAuley et al. The data is split as separate datasets according to the top-level product categories on Amazon. In our experiments, we use the Amazon Beauty (5-core) and Amazon Cell Phones and Accessories (5-core) (McAuley et al. 2015). We followed the common data preprocessing procedure from (Kang and McAuley 2018; Sun et al. 2019; Liu et al. 2021b). We treat each rating or review as a presence of a user-item interaction, group the interactions by user ids, and sort them by timestamps to form a sequence for each user. The detailed statistics of the datasets can be found in Table 1. Note that more than 75% sequence length are less than 9 and 7 for beauty and phones, respectively.

Experimental Settings

To evaluate all models, we adopted the widely used *leave-one-out* evaluation evaluation task (He et al. 2017; Kang and McAuley 2018; Sun et al. 2019; Liu et al. 2021b) to evaluate the sequential recommendation models. For each user, we use the last item of each use for testing, the second last item for validation, and the remaining items for training. Moreover, we follow the common practice in (Kang and McAuley 2018; Li, Wang, and McAuley 2020; Sun et al. 2019; Cho, Park, and Yoo 2020; Liu et al. 2021b) to pair each ground-truth item in the test set with 100 randomly sampled *negative* items that are not interacted by the user. We employ three evaluation metrics: *Recall* (HR), and *Normalized Discounted Cumulative Gain* (NDCG) and *Mean Reciprocal Rank* (MRR). The reported results are averaged for ten random seeds. Codes to reproduce the results are uploaded.

Baselines & Implementation Details

For the two transformer-based models SASRec (Kang and McAuley 2018) and BERT4Rec (Sun et al. 2019), and the two static models BPR-MF (Rendle et al. 2012) and LightGCN (He et al. 2020), we cite the experimental results from ASReP (Liu et al. 2021b). To implement AS-ReP and its re-training (RT) variants, we use code provided by the authors[2]. To ensure fairness, our method's implementation is based on their codes and only adds our

[1] https://jmcauley.ucsd.edu/data/amazon/

[2] https://github.com/DyGRec/ASReP

Table 2: Comparison of recommendation performance with baseline models in Beauty and Phones datasets based on Recall@5, NDCG@5, and MRR. Bold scores are the best in each row. Underlined scores are the second best.

| Model	Beauty			Phones		
	Recall@5	NDCG@5	MRR	Recall@5	NDCG@5	MRR
BPRMF	0.3737	0.2712	0.2682	0.3862	0.2849	0.2831
LightGCN	0.3852	0.2927	0.2906	0.4498	0.3394	0.3218
SASRec	0.3963	0.2949	0.2907	0.4646	0.3379	0.3314
BERT4Rec	0.4143	0.3128	0.3098	0.5077	0.3812	0.3720
RT-ASReP	0.4311	0.3257	0.3209	0.5213	0.3833	0.3667
ASReP	0.4684	0.3547	0.3458	0.5573	0.4193	0.4026
RT-BiCAT	_0.4769_	_0.3572_	_0.3470_	_0.5640_	_0.4211_	_0.4034_
BiCAT	**0.4901**	**0.3720**	**0.3618**	**0.5683**	**0.4290**	**0.4123**
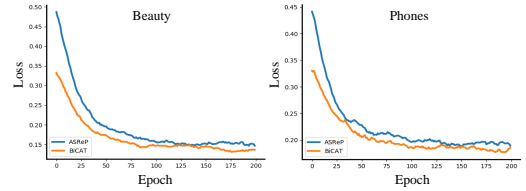
Figure 5: Fine-tuning on Beauty and Phones dataset respectively. The blue curves denote ASReP method, and orange curves denote our BiCAT method. In this plot, BiCAT have a great advantage to obtain better pre-train model and accelerate the convergence of network.

core component on it. Additionally, we also create the re-training (RT) variants of the BiCAT, which means training a new Transformer from scratch using the augmented sequences instead of fine-tuning the pre-trained Transformer. For all unique hyper-parameters and initialization strategies, we use the same setting of each model and report the results under each model's own optimal hyper-parameter setting. For reproducible checking, the hyperparameters of BiCAT are listed in the following. For beauty dataset, the best hyper-parameters are: max length $n = 100$, embedding size $d = 128$, the layer number $L = 2$, head number $h = 4$, dropout rate 0.7, ℓ_2 weight decay of $L2 = 0.0$, short sequence length threshold $M = 18$, and $K = 15$. For Phones dataset, $n = 100$, $d = 32$, $L = 2$, $h = 2$, dropout rate 0.5, $L2 = 0.0$, $M = 18$, and $K = 17$.

Table 3: Hyperparameter Analysis (Recall@5) over Beauty and Phones datasets. Bold score indicates the highest one.

Datasets	K=5	K=9	K=13	K=15
Beauty	0.4497	0.4589	0.4693	**0.4901**
Phones	0.5219	0.5316	0.5521	0.5630

Datasets	M=6	M=10	M=14	M=18
Beauty	0.4293	0.4446	0.4559	**0.4901**
Phones	0.5200	0.5342	**0.5719**	0.5683

Table 4: Comparison of recommendation performance on short sequences ($L \leq 3$) with baseline models in comprehensive evaluation metrics over Beauty and Phones dataset. Bold scores are the best in each row. Underlined scores are the second best.

Model	Dataset	Recall@1	Recall@5	Recall@10	NDCG@5	NDCG@10	MRR
SASRec	Beauty	0.1585	0.3623	0.4638	0.2651	0.2980	0.2641
	Phones	0.1935	0.4619	0.5889	0.3339	0.3750	0.3237
ASReP	Beauty	0.2185	0.4586	0.5698	0.3445	0.3804	0.3363
	Phones	0.2719	0.5687	0.6847	0.4278	0.4653	0.4089
BiCAT	Beauty	**0.2424**	**0.5018**	**0.6206**	**0.3787**	**0.4171**	**0.3676**
	Phones	**0.2830**	**0.5869**	**0.7103**	**0.4427**	**0.4828**	**0.4236**
Improv. vs. SASRec	Beauty	+52.93%	+60.05%	+33.81%	+42.85%	+39.97%	+39.19%
	Phones	+46.25%	+27.06%	+20.61%	+32.58%	+28.74%	+30.86%
Improv. vs. ASReP	Beauty	+10.94%	+9.42%	+8.92%	+9.93%	+9.65%	+9.31%
	Phones	+4.08%	+3.20%	+3.74%	+3.48%	+3.76%	+3.60%

Overall Performance Comparison (RQ1)

Table 2 shows the overall recommendation performance of our model and baselines on two datasets. We make two observations of overall comparison and ablation study:

- *Overall Comparison.* (1) The transformer-based models outperform the static models on all metrics. (2) However, they are worse than augmented models ASReP and BiCAT as they suffer from the cold-start problem for short sequences. Compared with SASRec, BiCAT has significant relative improvements on all three metrics by 24.77% on average. (3) Moreover, we observe that BiCAT outperforms ASReP on all three metrics by 3.45% on average. Note that the comparison is for sequences of all lengths. For long sequences with length over threshold M, we do not augment them with pseudo-prior items.

- *Ablation Comparison with re-train variants.* (1) The two variants RT-ASReP and RT-BiCAT are worse than AS-ReP and BiCAT, which verifies that pre-training is useful for sequential recommendations. (2) RT-BiCAT and Bi-CAT outperform ASReP in all datasets on all evaluation metrics. The improvement is based on the learning of bi-chronological item correlation to generate higher-quality pseudo-prior items for forward recommendation task.

Hyperparameter Study (RQ2)

To analyze the effect of the two core hyper-parameters K and M, which indicate the number of pseudo-prior items and the threshold for short sequences respectively, we report the performance results in Table 3. Note that, for each result, we only change the current one hyper-parameter but keep others the same as the optimal settings. We find that the performance generally improves on both datasets when K and M increase. It verifies the efficacy of augmenting pseudo-prior items for short sequences. Moreover, for Phones dataset, it shows that our BiCAT $M = 14$ is better than $M = 18$, the optimal setting in ASReP. For a direct comparison that lines with baselines, we still report the $M = 18$ results of BiCAT on Phones instead of the better $M = 14$ in Table 2.

Performance on Short Sequences (RQ3)

In this section, we conduct a study to discuss the effectiveness of BiCAT on short sequences and report the results in Table 4. To sum up, BiCAT outperforms other baselines on all six metrics. For short sequence with lengths smaller than 3 on both datasets, our BiCAT improves SASRec by an average increase of 40.12% on recall rates and ASReP by an average increase of 6.71% on recall rates. These results demonstrate that data augmentation of pseudo-prior items can be a promising method to alleviate the deficiency of short sequences learning. More importantly, the results verify that our method can generate higher-quality pseudo-prior items to improve recommendation performance.

Training Efficiency w.r.t Fine-tuning (RQ4)

In this section, to further verify our motivation that BiCAT can bridge the gap from augmentation to recommendation, we visualize the fune-tuning training loss. As shown in Figure 5, our pre-trained BiCAT brings the significant optimal initialized network parameters to facilitate the training of recommendation. We follow the same fune-tuning design in (Liu et al. 2021b) to make a fair comparison. As a common data augmentation method, our BiCAT can be applied in other models, not limited to sequential recommendations.

Conclusion

In this paper, we propose a state-of-the-art model for the learning of short sequences in sequential recommendation. Data augmentation of pseudo-prior items as a useful method can enhance the learning of short sequences and address the cold-start problem. We prove theoretically that the reverse augmentation ASReP may not be consistent with the forward sequential correlation. To generate higher-quality items, we propose BiCAT that uses a forward learning constraint in the loss to capture the contextual information when generating items reversely. Since the forward constraint is in line with the sequential recommendation task, it can bridge the gap between reverse augmentation and forward recommendation. Extensive experiments show the superiority of BiCAT by significant improvement over the state-of-the-art ASReP and SASRec. Moreover, our BiCAT can conduct

a better pre-trained Transformer to help main task quickly converge at the fine-tuning stage. For future studies, the generalization of BiCAT may also apply to other tasks such as video and language generation. How to generate dynamic items with respect to timestamps also pends future works.

References

Cho, S. M.; Park, E.; and Yoo, S. 2020. MEANTIME: Mixture of Attention Mechanisms with Multi-temporal Embeddings for Sequential Recommendation. In *Fourteenth ACM Conference on Recommender Systems*, 515–520.

Dong, M.; Yuan, F.; Yao, L.; Xu, X.; and Zhu, L. 2020. Mamo: Memory-augmented meta-optimization for cold-start recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 688–697.

Donkers, T.; Loepp, B.; and Ziegler, J. 2017. Sequential user-based recurrent neural network recommendations. In *Proceedings of the eleventh ACM conference on recommender systems*, 152–160.

He, R.; and McAuley, J. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 191–200. IEEE.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 639–648.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, 173–182.

Hidasi, B.; and Karatzoglou, A. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*, 843–852.

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

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Ji, M.; Joo, W.; Song, K.; Kim, Y.-Y.; and Moon, I.-C. 2020. Sequential recommendation with relation-aware kernelized self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 4304–4311.

Kang, K.; Park, J.; Kim, W.; Choe, H.; and Choo, J. 2019a. Recommender system using sequential and global preference via attention mechanism and topic modeling. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 1543–1552.

Kang, S.; Hwang, J.; Lee, D.; and Yu, H. 2019b. Semi-supervised learning for cross-domain recommendation to cold-start users. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1563–1572.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, 197–206. IEEE.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37.

Li, J.; Jing, M.; Lu, K.; Zhu, L.; Yang, Y.; and Huang, Z. 2019a. From zero-shot learning to cold-start recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4189–4196.

Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; and Ma, J. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1419–1428.

Li, J.; Wang, Y.; and McAuley, J. 2020. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*, 322–330.

Li, Y.; Luo, Y.; Zhang, Z.; Sadiq, S.; and Cui, P. 2019b. Context-aware attention-based data augmentation for poi recommendation. In *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*, 177–184. IEEE.

Liu, Q.; Wu, S.; Wang, D.; Li, Z.; and Wang, L. 2016. Context-aware sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 1053–1058. IEEE.

Liu, Z.; Chen, Y.; Li, J.; Yu, P. S.; McAuley, J.; and Xiong, C. 2021a. Contrastive Self-supervised Sequential Recommendation with Robust Augmentation. *arXiv preprint arXiv:2108.06479*.

Liu, Z.; Fan, Z.; Wang, Y.; and Yu, P. S. 2021b. Augmenting Sequential Recommendation with Pseudo-Prior Items via Reversely Pre-training Transformer. *arXiv preprint arXiv:2105.00522*.

Luo, Y.; Liu, Q.; and Liu, Z. 2021. STAN: Spatio-Temporal Attention Network for Next Location Recommendation. In *Proceedings of the Web Conference 2021*, 2177–2185.

Ma, C.; Kang, P.; and Liu, X. 2019. Hierarchical gating networks for sequential recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 825–833.

Ma, C.; Ma, L.; Zhang, Y.; Sun, J.; Liu, X.; and Coates, M. 2020a. Memory augmented graph neural networks for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 5045–5052.

Ma, J.; Zhou, C.; Yang, H.; Cui, P.; Wang, X.; and Zhu, W. 2020b. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 483–491.

McAuley, J.; Targett, C.; Shi, Q.; and Van Den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 43–52.

Quadrana, M.; Karatzoglou, A.; Hidasi, B.; and Cremonesi, P. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 130–137.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, 811–820.

Saveski, M.; and Mantrach, A. 2014. Item cold-start recommendations: learning local collective embeddings. In *Proceedings of the 8th ACM Conference on Recommender systems*, 89–96.

Smirnova, E.; and Vasile, F. 2017. Contextual sequence modeling for recommendation with recurrent neural networks. In *Proceedings of the 2nd workshop on deep learning for recommender systems*, 2–9.

Song, K.; Ji, M.; Park, S.; and Moon, I.-C. 2019. Hierarchical context enabled recurrent neural network for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4983–4991.

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 *Proceedings of the 28th ACM international conference on information and knowledge management*, 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 *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 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. In *Advances in neural information processing systems*, 5998–6008.

Wang, J.; Ding, K.; and Caverlee, J. 2021. Sequential Recommendation for Cold-start Users with Meta Transitional Learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1783–1787.

Wang, Q.; Yin, H.; Wang, H.; Nguyen, Q. V. H.; Huang, Z.; and Cui, L. 2019. Enhancing collaborative filtering with generative augmentation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 548–556.

Wang, Z.; Zhang, J.; Xu, H.; Chen, X.; Zhang, Y.; Zhao, W. X.; and Wen, J.-R. 2021. Counterfactual Data-Augmented Sequential Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 347–356.

Wu, L.; Li, S.; Hsieh, C.-J.; and Sharpnack, J. 2020. SSE-PT: Sequential recommendation via personalized transformer. In *Fourteenth ACM Conference on Recommender Systems*, 328–337.

Xie, X.; Sun, F.; Liu, Z.; Wu, S.; Gao, J.; Ding, B.; and Cui, B. 2020. Contrastive Learning for Sequential Recommendation. *arXiv preprint arXiv:2010.14395*.

Xin, X.; Karatzoglou, A.; Arapakis, I.; and Jose, J. M. 2020. Self-supervised reinforcement learning for recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 931–940.

Yao, T.; Yi, X.; Cheng, D. Z.; Yu, F.; Chen, T.; Menon, A.; Hong, L.; Chi, E. H.; Tjoa, S.; Kang, J.; et al. 2020. Self-supervised Learning for Large-scale Item Recommendations. *arXiv preprint arXiv:2007.12865*.

Yu, F.; Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 729–732.

Yuan, F.; He, X.; Jiang, H.; Guo, G.; Xiong, J.; Xu, Z.; and Xiong, Y. 2020. Future data helps training: Modeling future contexts for session-based recommendation. In *Proceedings of The Web Conference 2020*, 303–313.

Yuan, F.; Karatzoglou, A.; Arapakis, I.; Jose, J. M.; and He, X. 2019. A simple convolutional generative network for next item recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 582–590.

Zheng, Y.; Liu, S.; Li, Z.; and Wu, S. 2021. Cold-start Sequential Recommendation via Meta Learner. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 4706–4713.

Zhou, K.; Wang, H.; Zhao, W. X.; Zhu, Y.; Wang, S.; Zhang, F.; Wang, Z.; and Wen, J.-R. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 1893–1902.