Sequential Recommendation with Multiple Contrast Signals

CHENYANG WANG, WEIZHI MA, CHONG CHEN, MIN ZHANG, YIQUN LIU, and SHAOPING MA, Tsinghua University, China

Sequential recommendation has become a trending research topic for its capability to capture dynamic user intents based on historical interaction sequence. To train a sequential recommendation model, it is a common practice to optimize the next-item recommendation task with a pairwise ranking loss. In this paper, we revisit this typical training method from the perspective of contrastive learning and find it can be taken as a specialized contrastive learning task conceptually and mathematically, named *context-target contrast*. Further, to leverage other self-supervised signals in user interaction sequences, we propose another contrastive learning task to encourage sequences after augmentation, as well as sequences with the same target item, to have similar representations, called *context-context contrast*. A general framework, ContraRec, is designed to unify the two kinds of contrast signals, leading to a holistic joint-learning framework for sequential recommendation with different contrastive learning tasks. Besides, various sequential recommendation methods (e.g., GRU4Rec, Caser, and BERT4Rec) can be easily integrated as the base sequence encoder in our ContraRec framework. Extensive experiments on three public datasets demonstrate that ContraRec achieves superior performance compared to state-of-the-art sequential recommendation methods.

CCS Concepts: • **Information systems → Recommender systems;**

Additional Key Words and Phrases: Recommender system, sequential recommendation, self-supervised learning, contrastive learning

ACM Reference format:
Chenyang Wang, Weizhi Ma, Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. Sequential Recommendation with Multiple Contrast Signals. *ACM Trans. Inf. Syst.* 41, 1, Article 11 (January 2023), 27 pages.
https://doi.org/10.1145/3522673

1 INTRODUCTION

Sequential behaviors in modern web applications play a crucial role in recommender systems, such as product viewing, website clicking, music listening, etc. As a result, sequential recommendation has attracted increasing attention recently, which aims to predict the next action based on recent interactions. Traditional researches utilize Markov Chain to model transitions between items

This work is supported by the Natural Science Foundation of China (Grant No. U21B2026, 62002191) and Tsinghua University Guoqiang Research Institute.

Authors' addresses: C. Wang, C. Chen, M. Zhang (corresponding author), Y. Liu, and S. Ma, Department of Computer Science and Technology, Institute for Artificial Intelligence, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China; emails: {wangcy18, cc17}@mails.tsinghua.edu.cn, {z-m, yiqunliu, msp}@tsinghua.edu.cn; W. Ma, Institute for AI Industry Research (AIR), Tsinghua University, Beijing 100084, China; email: mawz@tsinghua.edu.cn.

1046-8188/2023/01-ART11 $15.00
https://doi.org/10.1145/3522673

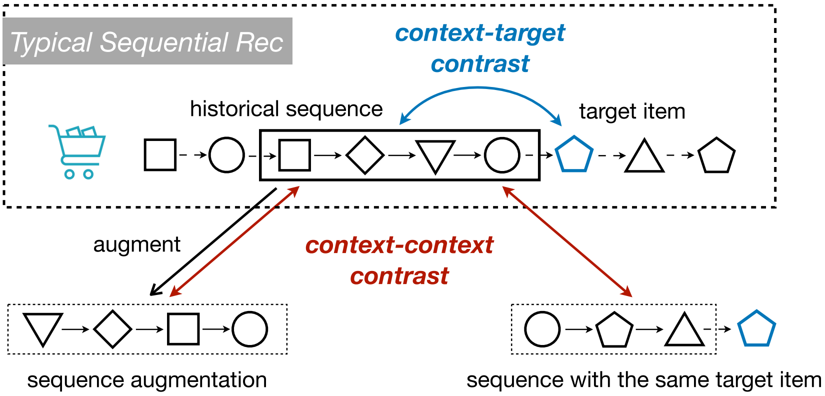


Fig. 1. Conceptual illustration of different contrast signals in sequential recommendation. The typical training method (next-item recommendation with pairwise ranking loss) can be seen as discriminating between the historical sequence and the target item (context-target contrast). Besides, we propose to leverage the contrast signal between historical sequences (context-context contrast). Intuitively, similar sequences (augmented sequences, sequences with the same target item) are expected to have similar representations.

[14, 15, 42, 51]. With the development of deep learning, recent efforts gain impressive progress on deep sequential recommendation models [18, 19, 24], where various neural networks are adopted as the sequence encoder to model dynamic user intents.

To train deep sequential recommendation models, many studies in this field follow a supervised learning paradigm, where each training instance contains an interacted item as the target and corresponding historical interactions as the input. However, different from standard supervised tasks in other domains, whose label information comes from external knowledge (e.g., category of an image) [4, 12, 13], the input and label here both intrinsically exist in the original interaction sequence. This makes it more consistent with the setting of self-supervised learning, which is a form of unsupervised learning that automatically generates supervision signals from data. From this perspective, we revisit the typical training method of sequential recommendation and propose that it actually acts as a specialized *contrastive learning* task.

Contrastive learning is a branch of self-supervised learning and aims at “learn to compare”, which constructs discrimination tasks from the data itself and has achieved impressive success in domains like **computer vision (CV)** and **natural language processing (NLP)** [33]. In sequential recommendation, the target item and historical interactions are both parts of the whole interaction sequence. Besides, given the historical sequence, the typical pairwise ranking objective [41] aims to drive the target item to get a higher ranking score than other items. Conceptually, this can be seen as discriminating between the historical sequence and the target item (called *context-target contrast*, shown in the upper part of Figure 1). Furthermore, we mathematically show that the pairwise ranking loss is a variant of the commonly adopted contrastive loss (details in Section 3.2).

Current studies mainly rely on such context-target contrast signal and aim to design powerful sequence encoders [19, 24, 31, 44, 61, 63]. Although this is capable of learning feasible sequence representations and has become the common practice, we argue that there are two primary problems. Firstly, the increasingly sophisticated sequence encoders (e.g., GRU [5], Transformer [47]) require a large amount of data and are prone to suffer from the data sparsity and overfitting issues. Secondly, most existing methods only focus on modeling correlations between the historical sequence and the target item, which ignore other contrast signals hidden in user interaction sequences.

To tackle the problems addressed above, we propose to additionally model the representation invariance between similar historical sequences to explore other contrast signals (called *context-context contrast*, shown in the lower part of Figure 1). Intuitively, sequences that reflect similar user intents should have similar representations. In this work, we investigate two kinds of "similar sequences" from the interaction data. For one thing, considering that a user's intent will not be influenced by small perturbations over the interaction sequence [45, 52], we devise several augmentation methods to construct different views of a given sequence as similar instances. For another, inspired by the idea of collaborative filtering [21], sequences followed by the same target item (even generated by different users) are likely to reflect similar user intents. As a result, we propose to encourage sequences after augmentation, as well as sequences with the same target item, to have close representations. We believe that incorporating such context-context contrast signal not only hinders sequential recommendation models from overfitting the next-item recommendation task, but also helps enhance sequence representations, and hence benefits the subsequent recommendation.

To unify the two kinds of contrast signals, namely context-target contrast and context-context contrast, we present a novel framework ContraRec, which jointly learns different contrastive learning tasks for sequential recommendation. For the context-target contrastive learning task, we extend the common BPR loss to a general contrastive loss by revisiting the typical training method from the perspective of contrastive learning. For the context-context contrastive learning task, a specific contrastive loss is designed to discriminate between similar and dissimilar interaction sequences, which supports multiple positive pairs compared to the common contrastive loss. Notably, ContraRec is a general framework that is flexible to integrate various sequence modeling methods as the base sequence encoder.

Extensive experiments on three public datasets show that ContraRec achieves superior performance compared to state-of-the-art sequential recommendation models. Additional experiments demonstrate the rationality of the proposed sequence augmentation methods. Besides, it is shown better to adopt a joint-learning framework rather than just utilizing the context-context contrast signal for pre-training and fine-tuning on the next-item recommendation task. ContraRec even outperforms the base sequential model with less than 40% of the training data, which exhibits its capability to make full use of the limited data.

The main contributions of this work are summarized as follows:

- We revisit the typical training method of sequential recommendation from the perspective of contrastive learning. The common BPR pairwise ranking loss is shown to be a specialized contrastive learning task conceptually and mathematically (called *context-target contrast* signal).
- We further explore additional contrast signals by modeling the representation invariance of similar historical sequences. Specifically, a generalized contrastive loss is devised to encourage sequences after augmentation, as well as sequences with the same target item, to have similar feature representations (called *context-context contrast* signal).
- A general framework, ContraRec, is presented to jointly learn the two kinds of contrast signals, leading to a holistic contrastive learning paradigm for sequential recommendation. ContraRec is also flexible to integrate various sequential recommendation models.
- Extensive experiments on three public datasets demonstrate that ContraRec leads to remarkable improvements compared to state-of-the-art sequential recommendation models, and the improvements are more significant when limited data is available.

The rest of this paper is organized as follows. We first introduce some preliminaries in Section 2. Next we revisit the typical training method of sequential recommendation and elaborate our

Table 1. Notations

Notation	Description
$\mathcal{U}$	the set of users
$\mathcal{I}$	the set of items
i_t	the target interacted item at time step t
S_t	the historical sequence before time step t
$\tilde{S}_t$	the augmented sequence of historical sequence S_t
$f(\cdot)$	the sequence encoder function
$g(\cdot)$	the similarity function
$\mathbf{i}_t \in \mathbb{R}^d$	the embedding of item i_t
$f(S_t) \in \mathbb{R}^d$	the representation of historical sequence S_t

ContraRec framework in Section 3. In Sections 4 and 5, we present the experimental results and related analyses to show the effectiveness and characteristics of ContraRec. Subsequently, some related work is reviewed in Section 6. We conclude this work and discuss the limitations as well as future directions in Section 7.

2 PRELIMINARIES

In this section, we first formulate the sequential recommendation problem and describe the typical training method (next-item recommendation with BPR pairwise ranking loss). Then we introduce the basic concepts of contrastive learning as well as the commonly adopted InfoNCE loss. Our main notations are summarized in Table 1.

2.1 Sequential Recommendation

Let $\mathcal{U}$ and $\mathcal{I}$ denote the user and item set, respectively. For each user $u \in \mathcal{U}$, we are given a chronologically ordered list $[i_1, i_2, \ldots, i_{N_u}]$, where each element $i_t \in \mathcal{I}$ is an item interacted at time step t and N_u is the length of the interaction sequence. Then the task of sequential recommendation is: given the historical sequence before the target time step t, denoted as S_t, generating an ordered list of items that the user may be interested in.

In general, most deep learning based sequential recommendation models focus on devising a specific sequence encoder $f(\cdot)$, which encodes the historical sequence S_t into a dense real-value vector $f(S_t)$ (called sequence representation). The sequence encoder generally converts items into embeddings first (the embedding of item i_t is denoted as $\mathbf{i}_t$). Then, various deep learning methods can be utilized to process the sequential data and generate the sequence representation, such as RNN [19], CNN [45], and attention mechanism [24]. Subsequently, a similarity function $g(\cdot)$ derives the ranking score $\hat{y}(S_t, i_t)$ between the historical sequence S_t and the target item i_t:

$$\hat{y}(S_t, i_t) = g\left(f(S_t), \mathbf{i}_t\right). \tag{1}$$

The similarity function usually takes a simple form and dot product is the most common practice, i.e., $g\left(f(S_t), \mathbf{i}_t\right) = \mathbf{i}_t^T f(S_t)$. As for the learning objective, most studies follow the next-item recommendation task and utilize BPR [41] pairwise ranking loss to train the model:

$$\mathcal{L}_{BPR} = \sum_{(S_t, i_t)} -\log \sigma\left(\hat{y}(S_t, i_t) - \hat{y}(S_t, i_t^-)\right), \tag{2}$$

where σ denotes the sigmoid function and i_t^- is a randomly sampled negative item that the user has not interacted with. This loss function aims to optimize the probability that the target item i_t gets a higher score than random negative items given the historical sequence S_t.

2.2 Contrastive Learning

Contrastive learning is a branch of self-supervised learning that aims to discriminate between similar and dissimilar samples directly generated from the data [33]. Different from generative self-supervised models (e.g., Autoencoder [27], Generative Adversarial Network [10]), contrastive methods do not need to recover "pixel-level" details of the data and at the same time retain necessary semantics via "learn to compare". The goal of contrastive learning is to encode data samples into a low-dimensional space so that samples that are semantically similar will be close to each other and far away from the others in the representation space.

Let x, y represent two similar data samples and $\mathbf{z}_x$, $\mathbf{z}_y$ denote their low-dimensional representation vectors. Besides, we have K negative samples that are dissimilar with x, whose representations are $[\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K]$. Generally, contrastive learning methods minimize the following temperature-scaled InfoNCE loss [38] as the contrastive loss:

$$l(\mathbf{z}_x, \mathbf{z}_y) = -\log \frac{\exp(g(\mathbf{z}_x, \mathbf{z}_y)/\tau)}{\exp(g(\mathbf{z}_x, \mathbf{z}_y)/\tau) + \sum_{k=1}^{K} \exp(g(\mathbf{z}_x, \mathbf{z}_k)/\tau)}, \tag{3}$$

where $g(\cdot)$ is a similarity function between representations (e.g., cosine similarity) and τ is a hyperparameter called temperature. Higher temperatures will lead to smoother distributions, and lower temperatures will make the loss focus more on adjacent samples. Optimizing this loss intuitively pushes up the similarity of semantically similar samples and pushes down the similarity when two samples are dissimilar.

3 THE PROPOSED FRAMEWORK

In this section, we first give an overview of the proposed ContraRec framework. Then, we elaborate two contrastive learning tasks in ContraRec and describe corresponding technical details. Finally, we discuss the characteristics of ContraRec in comparison to related methods.

3.1 ContraRec Overview

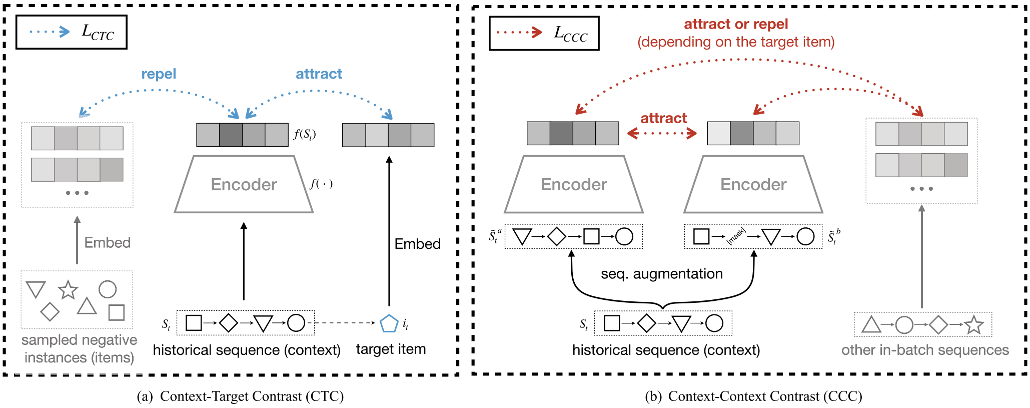
Figure 2 shows the overall structure of ContraRec. Different from the typical training method of sequential recommendation, ContraRec jointly learns two contrastive learning tasks: (1) **context-target contrast (CTC)** and (2) **context-context contrast (CCC)**.

On the one hand, we revisit the commonly adopted BPR pairwise ranking loss and find it actually acts as a specialized InfoNCE loss, which discriminates between historical sequences and target items. Based on this finding, we generalize BPR loss to context-target contrastive (CTC) loss, which is able to better support multiple negative samples during training.

On the other hand, to extract additional contrast signals in user interaction sequences, we propose to push up the representation similarity of similar historical sequences, leading to a context-context contrastive learning task. Here, similar sequences include not only different views generated from the same historical sequence but also sequences with the same next item. Finally, ContraRec unifies the two kinds of contrast signals with a joint learning framework.

3.2 Context-Target Contrast (CTC)

3.2.1 Revisiting Typical Training Method of Sequential Recommendation. Currently, most studies about sequential recommendation follow a supervised learning paradigm, where an interacted item is the prediction target, and its historical sequence serves as the input. Then, a BPR pairwise ranking loss as Equation (2) is generally optimized to train the sequential recommendation model. Here we show that such training strategy is actually a specialized contrastive learning task for two reasons, both conceptually and mathematically.



(a) Context-Target Contrast (CTC)

(b) Context-Context Contrast (CCC)

Fig. 2. Overview of the proposed ContraRec framework. ContraRec mainly consists of two contrastive learning task: (1) context-target contrast (CTC) and (2) context-context contrast (CCC). The first contrast signal (CTC) is a generalization of the common BPR pairwise ranking objective. As for the second contrast signal (CCC), we aim to push up the representation similarity of similar sequences. Here similar sequences include not only different views generated from the same historical sequence but also sequences with the same next item. Finally, ContraRec unifies the two kinds of contrast signals with a joint learning framework, leading to a holistic contrastive learning paradigm for sequential recommendation.

Firstly, different from supervised tasks in other domains, where the label information totally comes from external knowledge, the inputs and labels in sequential recommendation inherently exist in the original user interaction sequences. Current studies manually decouple the complete sequence into input historical sequences and target items to discriminate between them. Given the historical sequence, the target item is driven to get a higher ranking score compared to other items. This is conceptually consistent with the setting of contrastive learning: the historical sequence and the target item are similar samples directly extracted from the data, while other items are dissimilar instances given the historical sequence.

Secondly, the BPR pairwise ranking loss, i.e., Equation (2), can be proven a variant of contrastive loss as follows:

$$\begin{aligned}\mathcal{L}_{BPR} &= \sum_{(S_t,i_t)} -\log\sigma\left(\hat{y}(S_t,i_t) - \hat{y}(S_t,i_t^-)\right)\\ &= \sum_{(S_t,i_t)} -\log\left(\frac{1}{1+\exp\left(-\left(\hat{y}(S_t,i_t)-\hat{y}(S_t,i_t^-)\right)\right)}\right)\\ &= \sum_{(S_t,i_t)} -\log\left(\frac{\exp\left(\hat{y}(S_t,i_t)\right)}{\exp\left(\hat{y}(S_t,i_t)\right)+\exp\left(\hat{y}(S_t,i_t^-)\right)}\right).\end{aligned} \tag{4}$$

Remember that in most deep sequential recommendation models, $\hat{y}(S_t, i_t)$ is derived by a similarity function, i.e., Equation (1). Substituting the similarity function $g(\cdot)$ into Equation (4), we will get a similar format as the common contrastive loss, i.e., Equation (3). Specifically, this typical BPR pairwise ranking loss becomes a specialized InfoNCE loss $l(f(S_t), \mathbf{i}_t)$ under the following constraints: (1) there is only one negative sample in the denominator, and (2) the temperature $\tau = 1$. Therefore, the commonly adopted BPR loss in sequential recommendation is intrinsically related to the InfoNCE loss in contrastive learning. This typical training method actually discriminates between the representations of historical sequences and target items by optimizing a variant of contrastive loss.

3.2.2 Context-Target Contrastive Loss. Given the analyses above, the typical training method of sequential recommendation can be seen to discriminate between historical sequences and target items, named *context-target contrast* in this paper. Instead of directly optimizing BPR loss as the common practice, we further extend BPR to a general contrastive loss. Here we loosen the constraints summarized above and devise the following context-target contrastive (CTC) loss:

$$\mathcal{L}_{CTC} = \tau_1 \cdot \sum_{(S_t, i_t)} \overbrace{-\log\left(\frac{\exp\left(g(f(S_t), \mathbf{i}_t)/\tau_1\right)}{\exp\left(g(f(S_t), \mathbf{i}_t)/\tau_1\right) + \sum_{k=1}^{K} \exp\left(g(f(S_t), \mathbf{i}_k)/\tau_1\right)}\right)}^{l(f(S_t), \mathbf{i}_t)}, \tag{5}$$

where i_k is the sampled negative items that the user has not interacted with, and τ_1 is the temperature hyper-parameter. The outer coefficient τ_1 is to re-scale the gradient according to the derivation in previous work [11]. We use the common dot product as the similarity function $g(\cdot)$. When the number of negative items $K = 1$ and the temperature $\tau_1 = 1$, CTC loss degrades to BPR loss according to previous discussions.

Compared to the original BPR loss, CTC loss naturally supports multiple negative samples. Note that it is also possible to derive an enhanced version of BPR loss when there are multiple negative samples, named BPR+:

$$\mathcal{L}_{BPR+} = \sum_{(S_t, i_t)} -\frac{1}{K} \sum_{k}^{K} \log \sigma \left(\hat{y}(S_t, i_t) - \hat{y}(S_t, i_k)\right), \tag{6}$$

which repeats the positive instance multiple times to make it get a higher score than every negative item. However, this naive enhanced version does not consider the weights of different negative samples, in which case simple negative samples not only contribute little to the gradient updates but also weaken the contribution of hard negative samples due to the mean operation. On the contrary, the temperature hyper-parameter in CTC loss gives a simple but effective way to make the model pay different attentions to negative samples according to their hardness.

In our ContraRec framework, we adopt this CTC loss to learn the context-target contrast task. We will show its superiority compared to BPR and its enhanced version (BPR+) in Section 5.2, especially when the number of negative samples grows.

3.3 Context-Context Contrast (CCC)

In the section above, we show that the typical method to train sequential recommendation models can be taken as a contrastive learning task, which discriminates between historical sequences and target items extracted from the complete interaction sequence (*context-target contrast*). Current studies mainly rely on this task but neglect other self-supervision signals hidden in user interaction sequences. This motivates us to design other contrastive learning tasks so as to learn better sequence representations and overcome practical issues like data sparsity and overfitting.

In this section, we propose to additionally leverage the contrast signal between historical sequences (*context-context contrast*), which models the representation invariance of similar sequences. The basic idea is that historical sequences that reflect similar user intents should get close representations. Specifically, we first generate two augmented sequences (also called *views*) of the input historical sequence and assume that sequences after augmentation are intent-invariant. Subsequently, a specific sequence encoder derives low-dimensional representations of the augmented sequences, where various sequential recommendation methods can be adopted as the base sequence encoder. Then, a context-context contrastive (CCC) loss is proposed to encourage similar sequences to have close representations. Next, we introduce the three main components in detail.

3.3.1 Sequence Augmentation. The sequence augmentation component $Aug(\cdot)$ applies random augmentation to the original sequence. For each input historical sequence S_t, we generate two randomly augmented sequences $\tilde{S_t}^a = Aug(S_t, seed_1)$ and $\tilde{S_t}^b = Aug(S_t, seed_2)$. Here $seed_1$ and $seed_2$ are two random seeds that determine the concrete augmentation method (e.g., mask, reorder) and the effect of augmentation (e.g., masked positions). Given the augmentation set, $Aug(\cdot)$ first randomly chooses a specific augmentation method with equal probability, and then applies it on the input sequence. The design of augmentation methods in the augmentation set may relate to concrete application scenarios and recommendation models. In this work, we present two augmentation methods as examples, which will be detailed in Section 3.5.

3.3.2 Sequence Encoder. The sequence encoder component $f(\cdot)$ is responsible for encoding the input sequence S_t into a low-dimensional representation $f(S_t) \in \mathbb{R}^d$ (d is the dimension of the representation space). Note that the sequence encoder here is shared with that in the context-target contrastive learning task, i.e., there is only one set of parameters for the sequence encoder in our ContraRec framework. This makes the knowledge learned from different tasks able to benefit each other. As for the concrete architecture of the sequence encoder, there is no specific restriction. In practice, most deep learning based sequence modeling methods cater to the requirement, such as RNN, CNN, Transformer, and so on. This makes ContraRec a general framework, and various deep sequential recommendation models (e.g., GRU4Rec, Caser, BERT4Rec) can be integrated as the sequence encoder.

3.3.3 Context-Context Contrastive Loss. The context-context contrastive loss $\mathcal{L}_{CCC}$ aims to encourage the representations of similar sequences to be close to each other. Let us denote the original training mini-batch as $\mathcal{B} = \{(S_1, i_1), (S_2, i_2), \ldots\}$, whose element contains a target item i_t and corresponding historical sequence S_t. After the sequence augmentation component, we get an augmented sequence set $\mathcal{A} = \{\tilde{S_1}^a, \tilde{S_1}^b, \tilde{S_2}^a, \tilde{S_2}^b, \ldots\}$ with size $2|\mathcal{B}|$, where each original sequence yields two different views. Then we aim to find similar and dissimilar sequences within the augmented sequence set $\mathcal{A}$ (in-batch comparison).

For one thing, assuming the sequence augmentation can be seen as an intent-invariant transformation, the two sequences derived from the same historical sequence (e.g., $\tilde{S_1}^a$ and $\tilde{S_1}^b$) should have similar representations. For another, sequences derived from different historical sequences may also reflect similar user intents if their target items are the same (e.g., $\tilde{S_1}^a$ and $\tilde{S_2}^a$ assuming $i_1 = i_2$). Notice that for both circumstances, the identity of the target item[1] can be utilized as a sign of similar sequences. The augmented sequences also share the same target item because their original historical sequences are identical. Thus, we define the set of similar sequences of $\tilde{S_t}$ in $\mathcal{A}$ as $T(\tilde{S_t}) = \{\tilde{S_t'} \in \mathcal{A} \mid i_t = i_t'\}$. This set includes not only the paired augmented sequence of $\tilde{S_t}$, but also other augmented sequences that derived from different interaction sequences with the same target item (if exists). As a result, for each element $\tilde{S_t} \in \mathcal{A}$, there is possibly more than one positive instance in $\mathcal{A}$.

The common InfoNCE loss only supports one positive instance given an input instance. Thus, we extend InfoNCE loss and devise the following context-context contrastive loss $\mathcal{L}_{CCC}$ to encourage the representation of a given sequence $\tilde{S_t} \in \mathcal{A}$ to be close to the representations of all its similar

[1] Note that there is no information leakage here because the target items are visible during training.

ALGORITHM 1: Learning algorithm of ContraRec

Input: user-item interactions data $\bigcup_{u \in \mathcal{U}} S_u$; structure of sequence encoder $f(\cdot)$; sequence augmentation component $Aug(\cdot)$; CTC loss temperature τ_1; CCC loss temperature τ_2; CCC loss coefficient γ; number of negative samples K; batch size $|\mathcal{B}|$

Output: sequence encoder parameters Θ

```
Randomly initialize all parameters $\Theta$
for sampled mini-batch $\mathcal{B} = \{(S_t, i_t)\}$ from $\bigcup_{u \in \mathcal{U}} S^u$ do
    Sample $K$ negative items $\{i_k\}_{k=1}^{K}$ for each instance $(S_t, i_t)$
    Compute $\mathcal{L}_{CTC}$ according to Equation (5)
    $\tilde{S_t}^a = Aug(S_t, seed_1), \tilde{S_t}^b = Aug(S_t, seed_2)$        # sequence augmentation
    Compute $\mathcal{L}_{CCC}$ according to Equation (7)
    $\mathcal{L} \leftarrow \mathcal{L}_{CTC} + \gamma \mathcal{L}_{CCC} + \lambda ||\Theta||_2$
    Update model parameters $\Theta$ to minimize $\mathcal{L}$
end for
return $\Theta$
```

sequences $\tilde{S_t'} \in T(\tilde{S_t})$:

$$\mathcal{L}_{CCC} = \tau_2 \cdot \sum_{\tilde{S_t} \in \mathcal{A}} \frac{1}{|T(\tilde{S_t})|} \sum_{\tilde{S_t'} \in T(\tilde{S_t})} l\left(f(\tilde{S_t}), f(\tilde{S_t'})\right), \tag{7}$$

where $l(f(\tilde{S_t}), f(\tilde{S_t'}))$ is the contrastive loss between a single pair of similar sequence representations:

$$l\left(f(\tilde{S_t}), f(\tilde{S_t'})\right) = -\log \frac{\exp\left(\mathrm{sim}(f(\tilde{S_t}), f(\tilde{S_t'}))/\tau_2\right)}{\sum_{\tilde{S_t^-} \in \mathcal{A} \setminus \tilde{S_t}} \exp\left(\mathrm{sim}(f(\tilde{S_t}), f(\tilde{S_t^-}))/\tau_2\right)}. \tag{8}$$

The denominator in Equation (8) has $2|\mathcal{B}| - 1$ terms and cosine similarity is utilized to measure the difference between representations,[2] i.e., $g(\mathbf{x}, \mathbf{y}) = \cos(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y} / \|\mathbf{x}\| \|\mathbf{y}\|$. τ_2 is another temperature hyper-parameter as described in Section 2.2. It is noteworthy that $\mathcal{L}_{CCC}$ allows more than one similar sample in the mini-batch, which generalizes the commonly adopted InfoNCE loss in other domains.

3.4 Joint Learning

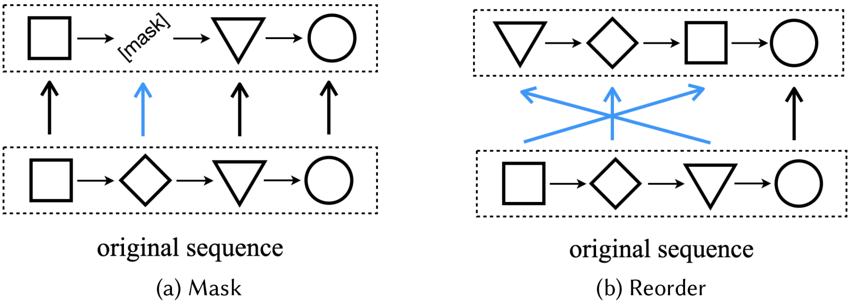
To unify the two kinds of contrastive learning tasks, namely context-target contrast and context-context contrast, we jointly optimize the two losses as a holistic contrastive learning framework:

$$\min_{\Theta} \quad \mathcal{L} = \mathcal{L}_{CTC} + \gamma \mathcal{L}_{CCC} + \lambda ||\Theta||_2, \tag{9}$$

where the context-context contrastive loss $\mathcal{L}_{CCC}$ is controlled by a coefficient γ and we add l2-normalization to all parameters. Note that the proposed ContraRec framework does not introduce any additional parameters, and the main parameters lie in the sequence encoder component. The overall learning procedure of ContraRec is demonstrated in Algorithm 1.

On the other hand, if the context-target contrast signal (next-item recommendation task in the typical training method) is taken as the main learning objective, the context-context contrast signal can be utilized for pre-training (optimizing $\mathcal{L}_{CCC}$ ahead of $\mathcal{L}_{CTC}$). However, the pre-training method isolates the two kinds of contrastive learning tasks and hinders model training from a

[2] We also tried dot product as the similarity function, but cosine similarity was shown to be more effective.



(a) Mask

(b) Reorder

Fig. 3. Illustration of the proposed sequence augmentation methods.

holistic contrastive learning framework. We will compare the performance of joint learning and pre-training in Section 4.3.

3.5 Sequence Augmentation Methods

Data augmentation methods are shown to be crucial for learning image representations in computer vision [22, 46]. However, in sequential recommendation, it remains unknown whether sequence augmentation would benefit the representation learning and how to augment historical sequences without changing the original user intents. As a primary attempt, we explore two sequence augmentation methods in this work, namely mask and reorder (shown in Figure 3). We believe that there exist other potential augmentation methods and different application scenarios may suit different augmentation or even combinations, which we plan to leave for future work.

3.5.1 Mask. In practice, users' intents are seldom dominated by a single interaction but generally remain stable over a period of time. A user may interact with a series of similar items with a specific intention. Based on this consideration, we propose to use random mask as a sequence augmentation method, which randomly masks a proportion of input tokens to a special mask token. This technique is also common in natural language processing tasks to avoid overfitting [2, 6, 55]. Specifically, the augmentation procedure can be formulated as follows:

$$\begin{aligned} &p_1 \sim \text{Beta}(\alpha = 3, \beta = 3), \quad n_1 = \lfloor N p_1 \rfloor, \\ &idx = \text{zeros\_like}(S_t), \\ &idx[: n_1] = 1, \quad \text{Shuffle}(idx), \\ &\tilde{S_t} = \text{Copy}(S_t), \quad \tilde{S_t}[idx] = [mask]. \end{aligned} \tag{10}$$

Given the historical sequence S_t (the length denoted as N), we first sample a proportion p_1 from a Beta distribution[3] and determine the number of positions n_1 we want to mask. Then, we randomly choose n_1 positions and set corresponding items in the sequence to the mask token.

3.5.2 Reorder. Intuitively, the order of user interactions in most real-world scenarios is in a flexible manner [45, 52]. Users may purchase a set of items simultaneously and the inner order does not matter. Besides, recent self-attention based sequential recommendation models gain significant performance improvements [24, 44] but care less about the temporal order compared to RNN, which implies that the sequence order may be not that important. Therefore, we propose to

[3] We also tried other parameter settings (α, β) of the Beta distribution and found the impacts are slight. Beta($\alpha = 3$, $\beta = 3$) usually yields promising results.

augment historical sequences based on partial permutation:

$$\begin{aligned} &p_2 \sim \mathrm{Beta}(\alpha = 3, \beta = 3), \quad n_2 = \lfloor N p_2 \rfloor, \\ &s_{start} \sim \{0, 1, 2, \ldots, N - n_2\}, \\ &s_{end} = s_{start} + n_2, \\ &\tilde{S_t} = \mathrm{Copy}(S_t), \quad \mathrm{Shuffle}(\tilde{S_t}[s_{start} : s_{end}]). \end{aligned} \tag{11}$$

Similarly, we sample a proportion p_2 first and determine the length n_2 we want to reorder. Subsequently, we uniformly select a continuous sub-sequence with length n_2 and randomly shuffle it, while interactions not included in the selected sub-sequence remain in the original order. This avoids shuffling the whole sequence each time and endows more randomness into the learning procedure, which potentially enhances the model robustness.

Both of the methods introduced above yield an augmented sequence $\tilde{S_t}$ with the same length N as the input sequence S_t. It is noteworthy that the sequence augmentation component is not restricted to the proposed two methods. We leave the investigation of other augmentation methods as future work to center our contribution to the overall contrastive learning framework.

3.6 Discussion

In summary, ContraRec presents a holistic contrastive learning framework for sequential recommendation, which consists of two kinds of contrastive learning tasks. For one thing, the commonly adopted next-item recommendation task with BPR pairwise ranking is revisited and generalized to a *context-target contrastive* (CTC) loss. For another, ContraRec models the representation invariance of similar historical sequences via the proposed *context-context contrastive* (CCC) loss, which encourages the sequences after augmentation, as well as sequences with the same target item, to have close representations. The two contrastive learning tasks are jointly optimized in the ContraRec framework. And various sequential recommendation models can be integrated to achieve better performance by learning better sequence representations. In this section, we compare ContraRec with two lines of related work, namely alternative training methods of sequential recommendation and contrastive learning in other domains.

3.6.1 Alternative Training Methods of Sequential Recommendation. Recent work [44, 58, 65] also begins to explore other self-supervision signals and alternative training strategies (e.g., masked item prediction, segment prediction, see Section 6 for more details). However, they mainly follow a two-stage learning paradigm, which first pre-train with other self-supervised tasks and fine-tune on the traditional next-item recommendation task. Neither of them investigates the connection between contrastive learning and next-item recommendation with BPR pairwise ranking loss. Differently, ContraRec makes the training of sequential recommendation a holistic contrastive learning framework together with the proposed context-context contrast signal. Besides, the self-supervised tasks in previous studies are mainly based on specific sequence encoders (e.g., Transformer block), while ContraRec does not introduce any extra parameters and serves as a general framework for various sequential recommendation models.

3.6.2 Contrastive Learning in Other Domains. Contrastive learning has achieved great success recently in domains like **computer vision (CV)** [4, 12, 37] and **natural language processing (NLP)** [9, 35, 55]. Take CV as an example, related studies usually transform input images into different views and utilize the InfoNCE loss to distinguish whether the augmented images come from the same input. However, in sequential recommendation, it is still not clear whether the data augmentation could benefit the representation learning and how to augment historical sequences without changing the original user intents. Hence, we devise two sequence augmentation methods in ContraRec to validate the effectiveness of data augmentation. Besides, considering the characteristics of sequential recommendation, the proposed $\mathcal{L}_{CCC}$ also encourages sequences with the same target item to have similar representations, which extends the common InfoNCE loss to support multiple positive pairs. Notice that if there is no historical sequence with the same target item in the mini-batch, $T(\tilde{S_t})$ has only one element (the paired augmented sequence of $\tilde{S_t}$) and $\mathcal{L}_{CCC}$ degrades to the commonly adopted InfoNCE loss in other domains.

4 EXPERIMENTS

4.1 Experimental Settings

4.1.1 Datasets. We conduct extensive experiments on three public datasets in real-world recommendation scenarios:

- *Beauty*[4]: This is one of the series of product review datasets crawled from Amazon [16]. The data is split into separate datasets by the top-level product category.
- *Yelp-2018*[5]: This is a popular dataset for business recommendation, including restaurants, bars and so on. We use the transaction records after *Jan. 1st, 2018* following previous work [53].
- *Gowalla*[6]: This is the check-in dataset [32] obtained from Gowalla, where users share their locations by checking-in.

To keep consistent with the default "5-core" setting in Amazon datasets, we preprocess the *Yelp-2018* and *Gowalla* dataset to ensure each user and item has at least five associated interactions. The statistics of datasets after preprocessing are summarized in Table 2.

4.1.2 Evaluation Protocols. We adopt the leave-one-out strategy to evaluate model performance, which is widely used in previous work [7, 49, 65]. For each interaction sequence, we use the most recent interaction for testing, the second recent interaction for validation, and the remaining interactions for training. In the meantime, considering that the *Gowalla* dataset contains repeat interactions for each user,[7] it is possible that the target item in the validation/test dataset has been seen in the training set. To avoid correlation between the three sets of data, we filter the validation and test data of *Gowalla* such that the target item is the first instance that the user has not been visited before (around 55% of the users meet the requirement). Some initial experiments were done without the filtering where the results were also positive, but in order to focus on avoiding the potential correlations between the data, only results from the filtered dataset will be presented.

To speed up model evaluation, we randomly sample 1,000 items as negative items, and this setting is shown to be close to the non-sampling version [28, 30]. We employ **Hit Ratio (HR)** and **Normalized Discounted Cumulative Gain (NDCG)** [23] as evaluation metrics. HR@k measures whether the target item appears in the top-k recommendation list, while NDCG@k further concerns about its position in the ranking list. Let $g_u \in [1, 1001]$ denote the rank of the target item for each user u, then HR@k and NDCG@k can be defined as follows under our experimental settings:

$$\begin{aligned} \text{HR@k} &= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} I(g_u \le k), \\ \text{NDCG@k} &= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{I(g_u \le k)}{\log_2(g_u + 1)}, \end{aligned} \tag{12}$$

[4] https://jmcauley.ucsd.edu/data/amazon/links.html.
[5] https://www.yelp.com/dataset.
[6] http://snap.stanford.edu/data/loc-gowalla.html.
[7] The *Beauty* and *Yelp-2018* datasets have been preprocessed to ensure that there is no repeat interaction.

Table 2. Statistics of Datasets After Preprocessing

Dataset	#user ($\|\mathcal{U}\|$)	#item ($\|\mathcal{I}\|$)	#inter ($\sum_u N_u$)	avg. length	density	time span
Beauty	22.4k	12.1k	198.5k	8.9	0.07%	2002.06.12 - 2014.07.23
Yelp-2018	88.4k	45.9k	1057.2k	12.0	0.03%	2018.01.01 - 2019.12.13
Gowalla	76.9k	304.4k	4616.6k	60.0	0.02%	2009.02.03 - 2010.10.22

where $I(\cdot)$ is an indicator function that only returns 1 when the condition is met, otherwise 0. We repeat each experiment five times with different random seeds and report the average score.

4.1.3 Baselines. We compare our method with various representative works, including state-of-the-art sequential recommendation models and recent studies based on self-supervised learning.

- **FPMC** [42]**:** This method combines matrix factorization and Markov Chains with a user-specific transition matrix.
- **GRU4Rec** [19]**:** This method utilizes **GRU (Gated Recurrent Unit**, a variant of RNN) [5] to model sequential interactions.
- **Caser** [45]**:** This is a CNN-based method that captures high-order Markov Chains by applying horizontal and vertical convolutional operations on interaction sequences.
- **SASRec** [24]**:** This method utilizes self-attention [47] to exploit the mutual influence between historical interactions.
- **BERT4Rec** [44]**:** This is a state-of-the-art sequential recommendation model that uses bidirectional self-attention [47] to derive sequence representations.
- **S^{3}Rec** [65]**:** It designs four pretext tasks for context-aware recommendation and then fine-tunes on the next-item recommendation task, which is a state-of-the-art method based on self-supervised learning. Due to the fact that there is no feature information in our setting, we only use the masked item prediction and segment prediction tasks in S^{3}Rec for fairness.
- **CLRec** [64]**:** This method uses the contrastive loss to optimize an attention-based sequential recommendation model to improve both effectiveness and fairness.
- **CL4SRec** [58]**:** This is a recent study that also investigates the contrast signal between augmented historical sequences based on contrastive learning. But it only takes sequences augmented from the same input as similar instances and uses a simple softmax loss.

4.1.4 Implementation Details. We implement our method with *PyTorch* and the codes are publicly available to facilitate reproducibility.[8] We use Adam [26] as the optimizer due to its success in recent deep learning methods. Early stop is adopted if NDCG@5 on the validation dataset continues to drop for 10 epochs. We consider a maximum of 50 recent interactions as the historical sequence.

We use the transformer layer in BERT4Rec as the default sequence encoder in our ContraRec because it has been shown to be effective in most scenarios. We will demonstrate the performance of ContraRec when integrating different base sequence encoders in Section 4.4. As for hyper-parameters of ContraRec, the coefficient of $\mathcal{L}_{CCC}$ is tuned within [0, 0.01, 0.1, 1, 5, 10]; the temperatures τ_1 and τ_2 are tuned from 0.1 to 1 with step 0.1 respectively; the batch size $|\mathcal{B}|$ is tuned between [256, 512, 1024, 2048, 4096]. With regard to the number of negative items in CTC loss, we set $K = 1$ by default for fair comparisons because baseline sequential recommendation models

[8] https://github.com/THUwangcy/ReChorus/tree/TOIS22.

generally use one negative sample during training. Besides, we will also report the results when $K = 64$ for ContraRec, denoted as ContraRec(multi). All the parameters are normally initialized with 0 mean and 0.01 standard deviation.

4.2 Overall Performance

For each dataset, we report the performances of two kinds of ContraRec according to whether there is only one negative sample during training:

- **ContraRec:** This is the proposed contrastive learning framework with $K = 1$ and $\tau_1 = 1$ in $\mathcal{L}_{CTC}$ (equivalent to jointly optimize $\mathcal{L}_{BPR}$ and $\mathcal{L}_{CCC}$). This ensures a fair comparison with base sequential recommendation models because they only use one negative item.
- **ContraRec(multi):** This is the full version of the proposed ContraRec, where the number of negative items K in $\mathcal{L}_{CTC}$ is set to 64 and τ_1 is tuned between 0.1 and 1.

Table 3 shows the performance and the running time of different methods. We summarize the experimental results in three folds.

4.2.1 Effectiveness of ContraRec. Firstly, self-supervised learning based methods are generally better than traditional sequential recommendation models, and CL4SRec becomes the strongest baseline. CL4SRec improves the performance of sequential recommendation models through encouraging the representation similarity between augmented sequences, which shows the usefulness of exploiting other contrast signals in sequential recommendation. Further, the proposed ContraRec and ContraRec(multi) achieve the best performance on all the datasets.

Compared to traditional sequential recommendation models, the main difference of ContraRec is optimizing the context-context contrastive loss in addition. The consistent improvements demonstrate that the context-context contrast signal between similar historical sequences indeed helps the model obtain better sequence representations and hence benefits the next-item recommendation task. Compared to S^{3}Rec and CLRec, the superiority of ContraRec shows that the sequence-augmentation based contrastive task is more beneficial to the representation learning in sequential recommendation. Compared to CL4SRec, we use the InfoNCE loss instead of the simple softmax loss, which is shown to be capable of better addressing the exposure bias in recommendation [64]. We also do not restrict the similar sequences to those augmented from the same input sequence, but incorporate sequences with the same target item inspired by the idea of collaborative filtering. This helps the proposed method achieve the best performance consistently.

4.2.2 Efficiency Analyses. Table 3 also shows the efficiency of different methods. To measure the running time, all the experiments are conducted with the same batch size of 256 and on the same machine (Intel Xeon 8-Core CPU of 2.2GHz and single NVIDIA GeForce RTX 2080Ti GPU) for fair compar ison. We notice that, in general, more recent models require more time for a single iteration (time/iter), while the convergence speeds (#iter) of different methods are usually diverse, leading to varying training time in total (total time). On the one hand, regarding the training time for a single iteration, our method is inevitably slower (no more 2×) than traditional sequential recommendation models because we need to encode additional two augmented sequences in each batch. Similarly, the recent contrastive method CL4SRec also needs more time for a single iteration. On the other hand, ContraRec converges very fast and only requires about 50 iterations to achieve stable performance, while CL4SRec usually needs more than 100 iterations. The reason is that our ContraRec leverages multiple contrast signals and uses the temperature-scaled contrastive loss to help the model focus more on hard negatives. As a result, the total training time of ContraRec is similar with traditional sequential methods (ContraRec and SASRec both need around 9.3h to achieve the optimal performance on the largest *Gowalla* dataset), which does not introduce

Table 3. Performance on Three Datasets

Method	*Beauty*						
	NDCG@5	HR@5	NDCG@10	HR@10	time/iter	#iter	total time
FPMC [41]	0.0933	0.1345	0.1112	0.1900	4.4s	∼100	∼7m
GRU4Rec [19]	0.0748	0.1113	0.0903	0.1591	8.1s	∼150	∼20m
Caser [45]	0.0662	0.1011	0.0814	0.1484	9.0s	∼150	∼23m
SASRec [41]	0.1070	0.1499	0.1245	0.2045	11.6s	∼100	∼20m
BERT4Rec [44]	0.1024	0.1453	0.1208	0.2022	13.1s	∼150	∼33m
S^{3}Rec [65]	0.1020	0.1426	0.1184	0.1934	12.1s	∼150	∼30m
CLRec [64]	0.1077	0.1536	0.1251	0.2071	12.3s	∼50	∼10m
CL4SRec [58]	0.1103	0.1567	0.1301	0.2215	20.9s	∼100	∼35m
ContraRec	0.1202**	0.1719**	0.1417**	0.2386**	20.2s	∼50	∼17m
ContraRec(multi)	**0.1300****	**0.1802****	**0.1496****	**0.2409****	22.1s	∼50	∼18m

Method	*Yelp-2018*						
	NDCG@5	HR@5	NDCG@10	HR@10	time/iter	#iter	total time
FPMC [41]	0.1395	0.2073	0.1719	0.4370	30.2s	∼150	∼1.3h
GRU4Rec [19]	0.1732	0.2540	0.2092	0.3659	44.1s	∼200	∼2.5h
Caser [45]	0.1549	0.2294	0.1898	0.3377	50.8s	∼200	∼2.8h
SASRec [41]	0.1842	0.2610	0.2189	0.3687	64.4s	∼100	∼1.8h
BERT4Rec [44]	0.1882	0.2668	0.2229	0.3743	72.5s	∼100	∼2.0h
S^{3}Rec [65]	0.1923	0.2703	0.2294	0.3801	65.2s	∼200	∼3.6h
CLRec [64]	0.2014	0.2901	0.2398	0.4012	71.3s	∼50	∼1.0h
CL4SRec [58]	0.2098	0.3002	0.2465	0.4103	127.1s	∼150	∼5.3h
ContraRec	0.2132*	0.3042*	0.2512*	0.4219**	121.2s	∼50	∼1.7h
ContraRec(multi)	**0.2193****	**0.3103****	**0.2568****	**0.4267****	144.2s	∼50	∼2.0h

Method	*Gowalla*						
	NDCG@5	HR@5	NDCG@10	HR@10	time/iter	#iter	total time
FPMC [41]	0.2498	0.3267	0.2752	0.4054	292.3s	∼100	∼8.1h
GRU4Rec [19]	0.2925	0.3959	0.3296	0.4108	299.8s	∼150	∼8.3h
Caser [45]	0.2736	0.3758	0.3119	0.4944	321.0s	∼100	∼8.9h
SASRec [41]	0.4742	0.6120	0.5124	0.7293	335.4s	∼100	∼9.3h
BERT4Rec [44]	0.4813	0.6215	0.5235	0.7324	377.5s	∼100	∼10.5h
S^{3}Rec [65]	0.4836	0.6302	0.5324	0.7398	340.1s	∼150	∼14.2h
CLRec [64]	0.4918	0.6362	0.5354	0.7458	378.1s	∼50	∼5.3h
CL4SRec [58]	0.5236	0.6646	0.5521	0.7725	680.1s	∼100	∼18.9h
ContraRec	0.5586**	**0.7067****	0.5913**	**0.8070****	667.3s	∼50	∼9.3h
ContraRec(multi)	**0.5662****	0.6976**	**0.5961****	0.7895**	780.2s	∼50	∼10.8h

The best result is in bold face, and the strongest baseline is underlined (** means significantly better than the strongest baseline with $p < 0.01$). "time/iter", "#iter", "total time" represents the training time for a single iteration, the number of iterations to converge, and the total training time, respectively (second/minute/hour [s/m/h]).

Table 4. Performance of Different Variants of ContraRec

Method	*Beauty*		*Yelp-2018*		*Gowalla*	
	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
BERT4Rec	0.1208	0.2022	0.2229	0.3743	0.5235	0.7324
Augment	0.1205	0.2027	0.2328	0.3884	0.5254	0.7395
ContraRec-Pre	0.1342	0.2301	0.2502	0.4103	0.5724	0.7826
ContraRec-SP	0.1381	0.2339	0.2508	0.4121	0.5875	0.7948
ContraRec-AP	0.1317	0.2256	0.2414	0.4013	0.5802	0.7893
ContraRec	**0.1417**	**0.2386**	**0.2512**	**0.4219**	**0.5913**	**0.8070**

Augment follows the typical BPR training paradigm but randomly augments the input sequence. ContraRec-Pre adopts the pre-training strategy. ContraRec-SP only encourages sequences augmented from the same input history to have similar representations. ContraRec-AP adds an additional linear projection to sequence representations before calculating $\mathcal{L}_{CCC}$.

significantly more time costs. Considering the remarkable improvements brought by ContraRec, we believe the performance gains somewhat justify the runtime costs.

4.2.3 Effect of More Negative Samples. Beyond the consistent improvements of ContraRec, we find ContraRec(multi) generally achieves higher performances compared to ContraRec. This indicates the usefulness of more negative samples during training, which brings richer contrast signals and further enhances the model performance. It is more possible to get hard negative items when the number of negative samples increases, hence providing a better estimation of the updating gradient. For general sequential recommendation models, although it is also feasible to use more than one negative sample with $\mathcal{L}_{BPR+}$ (i.e., Equation (6)), we will experimentally show in Section 5.2 that the proposed $\mathcal{L}_{CTC}$ benefits more from multiple negative samples. At the same time, experiments in the following sections will adopt the version when $K = 1$ (i.e., ContraRec instead of ContraRec(multi)) for fair comparisons with other methods, unless noted otherwise.

4.3 ContraRec Variants

To validate the plausibility of the proposed framework structure, we investigate the performance of four variants of ContraRec:

- **Augment:** This method follows the typical training paradigm but randomly augments the input historical sequence with our proposed augmentation component during training.
- **ContraRec-Pre:** This variant pre-trains the base sequential recommendation model with $\mathcal{L}_{CCC}$ for 50 epochs and then fine-tunes with $\mathcal{L}_{CTC}$.
- **ContraRec-SP:** This variant only encourages sequences augmented from the same input sequence to have close representations, in which case each sequence has a **single positive (SP)** instance, similar with the common practice of contrastive learning in other domains, i.e., $\mathcal{L}'_{CCC} = \sum_{S_t \in \mathcal{B}} l(f(\tilde{S_t}^a), f(\tilde{S_t}^b))$.
- **ContraRec-AP:** In other domains, it has been shown effective to add an **additional projection (AP)** head to the representation before calculating the contrastive loss [4]. This variant applies a learnable linear transform $\mathbf{M} \in \mathbb{R}^{d \times d}$ to sequence representations $f(\tilde{S_t})$ before calculating Equation (8), i.e., $l(\mathbf{M}^T f(\tilde{S_t}), \mathbf{M}^T f(\tilde{S_t'}))$.

Table 4 shows the performance of different methods on each dataset. We summarize the observations in four folds.

4.3.1 Effect of Sequence Augmentation. It is noteworthy that the Augment version usually performs a little better than the base sequential recommendation model BERT4Rec. Compared to the base sequential model, the only difference of the Augment version is randomly augmenting the input historical sequence during training. If the augmentation methods actually change the original user intents, the performance of the Augment version is expected to somewhat suffer a loss. On the contrary, the consistent performance improvements of the Augment version imply the rationality of the proposed sequence augmentation methods, namely mask and reorder. On the one hand, user intents are generally stable over a period of time, which will not be greatly influenced if some interactions are missing. On the other hand, user behaviors in many real-world scenarios are generally flexible and do not follow a rigid order.

4.3.2 Comparison of Training Strategy. Different from the joint learning framework of ContraRec, previous work [4, 65] usually adopts the two-stage learning strategy, which first pre-trains model parameters with pretext tasks and then fine-tunes on the main task. Here we treat context-context contrast ($\mathcal{L}_{CCC}$) as the pretext task and context-target contrast ($\mathcal{L}_{CTC}$) as the main task. From Table 4, it can be found that ContraRec-Pre also gets consistently better results compared to the base sequential recommendation model. This directly validates the usefulness of the proposed context-context contrastive learning task because it is only used for pre-training. The fine-tuning stage is the same as the training procedure of the base sequential recommendation model. Meanwhile, ContraRec-Pre is still inferior to ContraRec all the time. This suggests that it is better to adopt a holistic contrastive learning framework with joint learning but not just utilize the context-context contrast signal for pre-training. The two kinds of contrastive learning tasks ($\mathcal{L}_{CTC}$ and $\mathcal{L}_{CCC}$) potentially benefit each other during training, which is more suitable to be unified together.

4.3.3 Effect of Multiple Positive Pairs. Experiments demonstrate that ContraRec-SP suffers consistent performance losses, which implies the usefulness of multiple positive pairs in context-context contrast. This also differs our ContraRec from CL4SRec and contrastive learning studies in other domains. Although sometimes historical sequences with the same target item might reflect dissimilar user preferences (attracted by different aspects of the item), there are also many cases where the same target item indeed reflects similar user intents. As a result, pushing up the similarity of their representations helps to enhance the capacity of the sequence encoder. Notice that the performance drop of ContraRec-SP is not that large. One possible reason is that some mini-batches may not contain historical sequences with the same target item because there are numerous items in the dataset. Even though, we argue this is a unique supervision signal in sequential recommendation that should be taken into consideration, which enriches the context-context contrast signal and further boosts the recommendation performance.

4.3.4 Effect of Additional Projection Head. Different from the observation in other domains, we find adding a projection head hurts the overall performance of ContraRec. Although ContraRec-AP still outperforms the base sequential recommendation model, it is inferior to ContraRec all the time. This may result from the ranker function in sequential recommendation, which generally takes a simple form like dot product. Unlike models in computer vision that needs another linear classification layer after getting the image representation, sequential recommendation directly recommends items close to the sequence representation. Hence, it may be better to drive the representations of similar sequences to be close to each other in the original representation space.

4.4 Integration with Different Base Sequence Encoders

Note that the proposed ContraRec serves as a general framework and different sequence models can be easily integrated as the base sequence encoder. In the main experiments, we adopt BERT4Rec as the default encoder for the effectiveness of the transformer layer. To validate the

Table 5. Performance of ContraRec when Adopting Different Base Sequence Encoders

Method	*Beauty*		*Yelp-2018*		*Gowalla*	
	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
GRU4Rec	0.0903	0.1591	0.2092	0.3659	0.3296	0.5108
+ ContraRec	0.1273	0.2177	0.2371	0.3920	0.4245	0.6268
Caser	0.0814	0.1484	0.1898	0.3377	0.3119	0.4944
+ ContraRec	0.1063	0.1864	0.2084	0.3570	0.4344	0.6381
BERT4Rec	0.1208	0.2022	0.2229	0.3743	0.5235	0.7324
+ ContraRec	0.1417	0.2386	0.2512	0.4219	0.5913	0.8070

flexibility of our ContraRec, Table 5 shows the results of integrating GRU4Rec, Caser, and BERT4Rec (three representative deep learning techniques to model sequential data, i.e., RNN, CNN, and Transformer) as the base sequence encoder, respectively. We can see that ContraRec brings significant performance improvements to distinct base sequential models on all the datasets. This shows the effectiveness and flexibility of our framework. It is noteworthy that GRU4Rec+ContraRec even outperforms BERT4Rec on *Yelp-2018* with a less expressive sequence encoder. Meanwhile, the best performing BERT4Rec also gets encouraging improvements with ContraRec.

4.5 Comparison of Different Augmentation Methods

To validate the effectiveness of the two proposed sequence augmentation methods, four augmentation strategies are compared in this section. Remember that we randomly choose a specific augmentation method with equal probabilities by default in ContraRec, denoted as **Random Choice**. Here, we test the performance when only one augmentation method is adopted, denoted as **Mask Only** and **Reorder Only**, respectively. Besides, we also compare the case when two augmentation methods are simultaneously applied on the input sequence, denoted as **Stacking**. Table 6 shows the performance of different augmentation methods on each dataset.

We observe that different augmentation methods perform differently and there does not exist a golden augmentation strategy. The best results are achieved by distinct strategies for different datasets. For example, Reorder Only performs the best on *Beauty* and *Yelp-2018*, while Random Choice is more suitable for *Gowalla*. This indicates that distinct augmentation methods may learn different knowledge, and hence are good at some certain scenarios. Designing other dataset-specific augmentation or exploring adaptive sequence augmentation methods is probably a promising future direction, which needs some domain-knowledge and a deep understanding of the data.

In the meantime, there are also some patterns that can be concluded from the experimental results. First, considering the single augmentation method, Reorder Only generally yields promising results while Mask Only performs a little worse. The reorder operation might introduce some knowledge that can be hardly learned by the traditional context-target contrastive task (i.e., BPR pairwise ranking), which implies that the temporal order of interactions is not that important in these datasets. However, we think this is possibly highly related to the concrete recommendation scenarios. For product review dataset *Beauty*, users' preferences might not change much over time, and the average length of the user interaction sequence is short. For *Gowalla*, the interactions are spaced more closely in time as shown in Table 2, and most users are only active within a shorter time period. While for other datasets that the time-series information is especially important, it will be better to investigate other data-specific augmentation methods. Second, although the mask

Table 6. Performance of Different Sequence Augmentation Strategies

Method	*Beauty*		*Yelp-2018*		*Gowalla*	
	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
Mask Only	0.1352	0.2275	0.2166	0.3942	0.5879	0.8038
Reorder Only	**0.1424**	**0.2400**	**0.2576**	**0.4273**	0.5913	0.8033
Stacking	0.1307	0.2216	0.2423	0.4021	0.5820	0.7964
Random Choice	0.1417	0.2386	0.2512	0.4219	**0.5913**	**0.8070**

Mask Only and Reorder Only adopt a single augmentation method respectively. Stacking simultaneously applies two augmentation methods on the input sequence. Random Choice selects a specific augmentation method with equal probabilities.

operation does not bring significant performance gains when adopted solely, Random Choice is superior to Reorder Only on *Gowalla* with the incorporation of the mask operation. This implies the potential importance to combine different augmentations in some certain datasets, which may especially benefit from encouraging sequences after different augmentations to have similar representations. Third, stacking multiple augmentation methods is not as useful as expected, which yields sub-optimal performance. Applying various augmentation methods together may change the structure of the original sequence to a large extent, in which case the user intent can hardly be ensured to stay consistent. Finally, although Random Choice is not always the best performing strategy, it generally achieves encouraging results. Thus, it is reasonable to choose Random Choice as the default augmentation strategy in ContraRec.

5 FURTHER ANALYSES

5.1 Training Data Size

One of the motivations of additionally exploring the context-context contrast signal in ContraRec is to alleviate the data sparsity issue. We first analyze the impacts of training data size and validate whether ContraRec is capable of making better use of the data. Figure 4 shows the performances of ContraRec and BERT4Rec when utilizing different ratios of training data on *Yelp-2018*. Results on other datasets and combinations with other sequence encoders are similar.

First, we can see different methods generally yield better results when more training data is available. This shows the importance of training data size for deep learning based methods. Second, ContraRec consistently outperforms the base sequential recommendation model when utilizing different ratios of training data. It is noteworthy that with only 40% of the data, ContraRec is superior to the full base model using all the data. Besides, the relative improvements of ContraRec are more obvious when limited data is available, and there are still encouraging performance gains with all the data. These observations imply that the the additional self-supervised signal (i.e., context-context contrast) in ContraRec helps to make full use of the data and is potentially capable of alleviating the data sparsity problem in practice.

5.2 Number of Negative Samples in CTC Loss

In Section 3.2, we revisit the typical pairwise ranking training method and extend BPR loss to a general contrastive loss $\mathcal{L}_{CTC}$, which supports multiple negative samples during training. Here, we investigate how does the performance change as the number of negative samples increases. Figure 5 shows the performance of ContraRec on *Beauty* when varying the number of negative samples. Results on other datasets demonstrate similar trends. For comparison, we also report the performances when substituting $\mathcal{L}_{CTC}$ with $\mathcal{L}_{BPR}$ (only one negative sample) and $\mathcal{L}_{BPR+}$ (i.e., Equation (6)), respectively. From the figure, we mainly have two observations as follows.

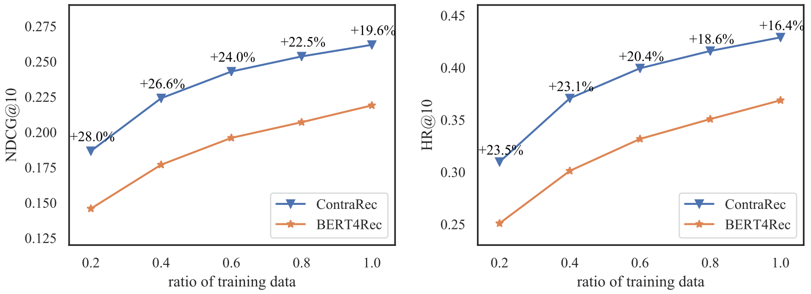


Fig. 4. Performance when using different ratios of training data on *Yelp-2018*. Relative improvements compared to the base model are annotated.

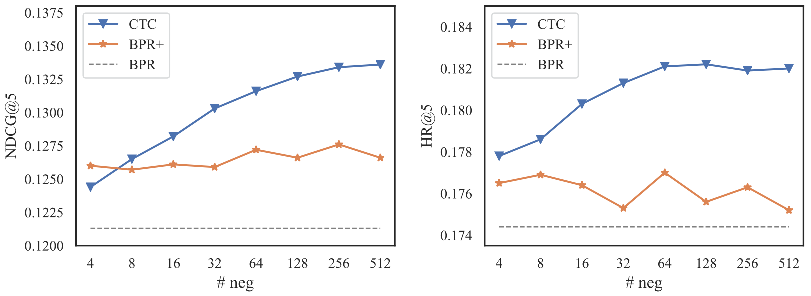


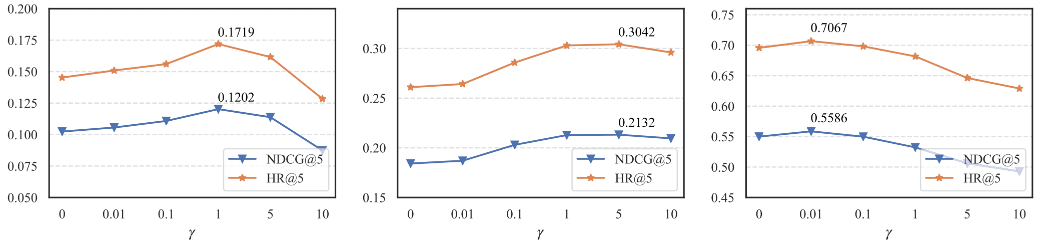
Fig. 5. Performance when varying the number of negative samples during training on *Beauty*. The proposed CTC loss benefits more from multiple negative samples than BPR+.

First, it is generally beneficial to use more negative samples. Notice that $\mathcal{L}_{CTC}$ and $\mathcal{L}_{BPR+}$ both outperform $\mathcal{L}_{BPR}$ (the dotted line) when there are multiple negative samples. This shows the usefulness of sampling multiple negative items, in which case hard negative samples are more probable to be included and provide more valuable gradient updates. Second, the proposed $\mathcal{L}_{CTC}$ benefits more from multiple negative samples. It is noteworthy that the performance when using $\mathcal{L}_{CTC}$ steadily rises up as the number of negative samples increases. As for $\mathcal{L}_{BPR+}$, although it gets slightly better results compared to $\mathcal{L}_{BPR}$, its performance does not benefit from more negative samples. This is mainly because $\mathcal{L}_{BPR+}$ does not weigh each negative sample differently, in which case simple negative samples do not provide useful gradient updates and weaken the contributions of hard negative samples at the same time. On the contrary, the temperature hyper-parameter in $\mathcal{L}_{CTC}$ provides a simple but effective way to control the model's attention payed to different negative samples. Therefore, we propose that $\mathcal{L}_{CTC}$ can become an optional substitute for $\mathcal{L}_{BPR}$ to train general sequential recommendation models in practice.

5.3 Parameter Sensitivity

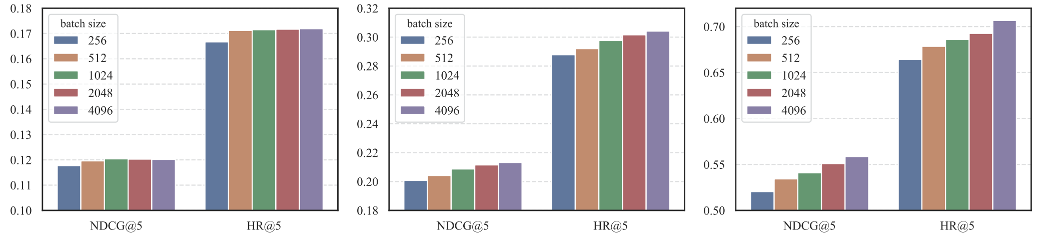
Then, we conduct a series of experiments to investigate the impacts of major hyper-parameters in our ContraRec framework, including the coefficient of the context-context contrastive loss γ, the batch size $|\mathcal{B}|$, and the temperature τ_2 in the context-context contrastive (CCC) loss.

5.3.1 Coefficient of the Context-Context Contrastive Loss. The coefficient γ in the final objective function Equation (9) controls the importance of the context-context contrastive learning task. Figure 6 shows the NDCG@10 of ContraRec with different base sequential recommendation models when the coefficient ranges within [0, 0.01, 0.1, 1, 5, 10]. First, it can be observed that the



(a) Beauty (b) Yelp-2018 (c) Gowalla

Fig. 6. Effect of the coeffecient γ of the context-context contrastive loss $\mathcal{L}_{CCC}$. The performance generally increases first, and then decreases.



(a) Beauty (b) Yelp-2018 (c) Gowalla

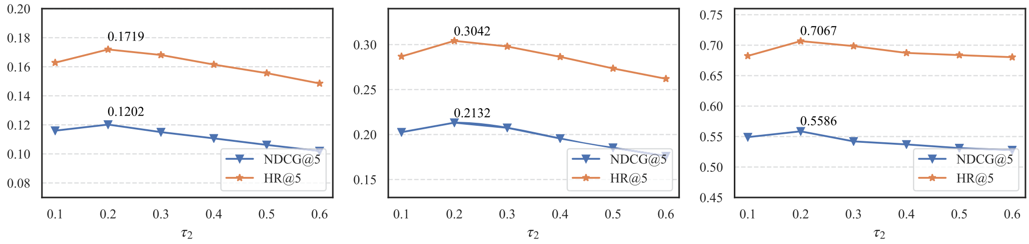
Fig. 7. Effect of the batch size during training. ContraRec usually benefits from a larger batch size.

context-context contrastive learning task indeed benefits the recommendation performance. It is useful to encourage sequences after augmentation as well as sequences with the same target item to have similar representations. Although the typical context-target contrastive task can also achieve the goal that multiple sequences with the same target learn similar embeddings, the CCC loss further addresses this explicitly. Besides, the knowledge to learn augmentation-invariant representations in the CCC loss can hardly be captured by the traditional learning paradigm. Compared to the situation when only optimizing $\mathcal{L}_{CTC}$ ($\gamma = 0$), ContraRec achieves better results under most settings of γ on all the datasets. Besides, the overall trend increases first and then decreases. Notice that $\gamma = 1$ generally yields promising results, while the best setting varies across datasets, which may rely on the data scale and the concrete scenario.

5.3.2 Batch Size. Considering the context-context contrastive learning task in ContraRec, the number of negative samples in $\mathcal{L}_{CCC}$ is directly relevant to the training batch size due to in-batch comparison. According to [38], the negative InfoNCE loss will be a tighter lower bound of mutual information between similar instances when the number of negative samples increases. Besides, a larger batch size will make it more possible that different sequences in the batch share the same target item. Therefore, large batch sizes are expected to lead to better results. Figure 7 shows the performance when different batch sizes are adopted. It can be seen that ContraRec indeed benefits from larger batch sizes. The best result is generally achieved with the batch size of 4,096,[9] while ContraRec is still much better than the base sequential recommendation model with a small batch size of 256.

5.3.3 Temperature. Another important hyper-parameter is the temperature τ_2 in Equation (8), which controls the smoothness of the softmax distribution. Figure 8 shows the performance of ContraRec when the temperature ranges from 0.1 to 0.6. We find most settings of τ_2 lead to better

[9] We did not try the batch size of 8,192 or higher because of the limited GPU memory.



(a) Beauty (b) Yelp-2018 (c) Gowalla

Fig. 8. Effect of the temperature in the context-context contrastive (CCC) loss.

results than the base sequential encoder, and a small temperature around 0.2 generally yields encouraging results, which is also stable across datasets. When the temperature is small, the model may benefit from focusing more on similar samples while neglecting the ones too far away. In the meantime, too small or especially too large temperatures will also hurt the performance.

6 RELATED WORK

In this section, we review two lines of work related to ours and present the main differences of the proposed ContraRec.

6.1 Sequential Recommendation

Different from general recommendation methods [3, 17, 41], sequential recommendation leverages historical sequences to better capture current user intent [39]. Previous studies depend on Markov chains to model the transition pattern between items [42, 43, 50]. Recently, with the development of deep learning, there has been much work utilizing deep models to encode historical sequences to hidden vectors [8, 19, 29, 48, 49, 61, 63]. GRU4Rec [19] first introduces RNN to the seqeuntial recommendation domain, and Caser [45] utilizes a CNN-based method to capture high-order Markov chains by applying convolutional operations on historical sequences. Besides, inspired by the effectiveness of the attention mechanism in other domains [1, 59], SASRec [24] first applies self-attention to model the mutual influence between historical interactions, achieving impressive performance improvements. However, most deep sequential recommendation models focus on the structure of the sequence encoder and rely on the next-item recommendation task (*context-target contrast*), while ignoring other self-supervised signals like *context-context contrast* addressed in ContraRec.

On the other hand, some recent work about sequential recommendation begins to explore alternative training strategies. BERT4Rec [44] adopts a Cloze objective that predicts the randomly masked items in the sequence by their context. Ma et al. [36] propose to predict the future sequence instead of next item and construct a seq2seq training strategy based on disentanglement. Yuan et al. [62] investigate the pretraining strategy on sequential data and transfer the learned parameters to downstream tasks. However, neither of them investigates the contrast signals directly extracted from historical sequences (e.g., finding similar sequences based on augmentation and target item in ContraRec). Besides, they usually need specific sequence encoders to fit their training strategies, while ContraRec does not introduce any extra parameters and serves as a general framework to boost various deep sequential recommendation models.

6.2 Contrastive Learning

Contrastive learning is a branch of self-supervised learning, which obtains supervision signals from the data itself and usually predicts part of the data from other parts [33]. Compared to supervised learning that needs manual labels, self-supervised learning has drawn increasing attention for its data efficiency and generalization ability. There are mainly two directions in self-supervised learning, namely generative methods and contrastive methods. Generative methods [7, 40] generally aim to reconstruct "pixel-level" details of the transformed data, while contrastive methods [4, 12, 20, 25, 38] discriminate the relationship of paired samples in a classification manner. Considering that a good representation may still not be powerful enough to recover details of the data, contrastive methods are more consistent with human intuition and have achieved great success in CV and NLP recently.

Deep InfoMax [20] first uses a contrastive learning task to model the mutual information between a local patch and its global context. Subsequently, CPC [38] pioneers the practice of maximizing mutual information between the past and future parts in sequential data. More recently, studies begin to directly discriminate between similar and dissimilar samples, where using rich negative samples is shown to be important [56]. MoCo [12] develops momentum contrast learning with a dynamic queue to increase the number of negative samples. SimCLR [4] further illustrates the importance of a hard positive sample strategy by data augmentation, which generates different views of an input image as similar samples and adopts in-batch comparison to obtain dissimilar samples.

Inspired by the success of contrastive learning in other domains, researchers in the recommendation domain also begin to investigate possible application directions. Yao et al. [60] devise a self-supervised learning framework based on feature correlations of items. Zhuang et al. [34] and Wu et al. [54] propose to apply contrastive learning on graph neural networks for graph-based recommender systems, which construct self-supervised signals through graph augmentation. In the domain of sequential recommendation, S^3-Rec [65] devises four self-supervised tasks from raw features of items in the interaction sequence. CLRec [64] uses the contrastive loss to address the exposure bias in recommendation. Xia et al. [57] try to construct hypergraph for session-based recommendation and leverage contrastive learning to improve the graph representation learning. Xie et al. [58] design a contrastive learning task based on sequence augmentation, which might be the most relevant work to us. However, they only consider the context-context contrast signal of augmented sequences from the same input sequence, and only use a simple softmax loss to model the next-item recommendation task. The proposed ContraRec framework differs from these studies by providing a holistic contrastive learning paradigm for sequential recommendation. On the one hand, we first find the typical BPR pairwise ranking loss is a specialization of InfoNCE loss and extend it to a general context-target contrastive loss. On the other hand, we creatively define two kinds of "similar sequences" based on sequence augmentation and the identity of target item, which is not fully explored in previous work. Besides, different from the typical pre-training strategy [58, 65], ContraRec jointly optimizes the two contrastive learning tasks, leading to a holistic contrastive learning framework for sequential recommendation.

7 CONCLUSION AND FUTURE WORK

In this work, we first revisit the typical training method of sequential recommendation (next-item ranking with a pairwise ranking loss) from the perspective of contrastive learning, which can be taken as a specialized contrastive learning task (called *context-target contrast*). Based on this finding, we extend the common BPR pairwise ranking loss to a general contrastive loss $\mathcal{L}_{CTC}$. Besides, we propose another contrastive learning task to explore other self-supervised signals hidden in user interaction sequences. Specifically, a specialized contrastive loss $\mathcal{L}_{CCC}$ is devised to model the representation invariance between similar interaction sequences, where sequences after augmentation, as well as sequences with the same target item, are encouraged to have similar representations (called *context-context contrast*).

Furthermore, we present a general framework ContraRec to unify the two kinds of contrast signals with joint learning, leading to a holistic contrastive learning framework for sequential recommendation. To generate different views of the input sequence, we adopt two sequence augmentation methods, namely mask and reorder. ContraRec is flexible to integrate various existing sequential recommendation models as the base sequence encoder. Extensive experiments on three public datasets demonstrate that ContraRec brings significant improvements, especially when limited data is available.

At the same time, our ContraRec framework still has some limitations. On the one hand, different recommendation scenarios may suit different sequence augmentation methods. We mainly give two example augmentation methods and they may not cater to the characteristics of other datasets (especially the reorder method). It will be interesting to design domain-specific augmentation or explore adaptive augmentation methods based on meta-learning. On the other hand, the current framework requires a large batch size to provide adequate negative samples, which may result in potential training issues with the limitation of computational resources. In the future, other contrastive learning structures can be investigated to alleviate this problem.

REFERENCES

[1] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

[2] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016*. Association for Computational Linguistics (ACL), 10–21.

[3] Chong Chen, Min Zhang, Chenyang Wang, Weizhi Ma, Minming Li, Yiqun Liu, and Shaoping Ma. 2019. An efficient adaptive transfer neural network for social-aware recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 225–234.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*. PMLR, 1597–1607.

[5] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1724–1734.

[6] Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. *Advances in Neural Information Processing Systems* 28 (2015), 3079–3087.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.

[8] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1555–1564.

[9] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821* (2021).

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[11] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision* 129, 6 (2021), 1789–1819.

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9729–9738.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[14] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2018. Translation-based recommendation: A scalable method for modeling sequential behavior. In *IJCAI*. 5264–5268.

[15] Ruining He and Julian McAuley. 2016. Fusing similarity models with Markov chains for sparse sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 191–200.

[16] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*. 507–517.

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

[18] Balázs Hidasi and Alexandros Karatzoglou. 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.

[19] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[20] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*.

[21] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 263–272.

[22] Paras Jain, Ajay Jain, Tianjun Zhang, Pieter Abbeel, Joseph E. Gonzalez, and Ion Stoica. 2020. Contrastive code representation learning. *arXiv preprint arXiv:2007.04973* (2020).

[23] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[24] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 197–206.

[25] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems* 33 (2020).

[26] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Computer Science* (2014).

[27] Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational Bayes. *Stat* 1050 (2014), 1.

[28] Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1748–1757.

[29] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 95–104.

[30] Dong Li, Ruoming Jin, Jing Gao, and Zhi Liu. 2020. On sampling Top-K recommendation evaluation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2114–2124.

[31] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[32] Dawen Liang, Laurent Charlin, James McInerney, and David M. Blei. 2016. Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web*. 951–961.

[33] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[34] Zhuang Liu, Yunpu Ma, Yuanxin Ouyang, and Zhang Xiong. 2021. Contrastive learning for recommender system. *arXiv preprint arXiv:2101.01317* (2021).

[35] Dongsheng Luo, Wei Cheng, Jingchao Ni, Wenchao Yu, Xuchao Zhang, Bo Zong, Yanchi Liu, Zhengzhang Chen, Dongjin Song, Haifeng Chen, et al. 2021. x Unsupervised document embedding via contrastive augmentation. *arXiv preprint arXiv:2103.14542* (2021).

[36] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 483–491.

[37] Ishan Misra and Laurens van der Maaten. 2020. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6707–6717.

[38] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[39] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems. *ACM Computing Surveys (CSUR)* 51, 4 (2018), 1–36.

[40] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. 2019. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*. 14866–14876.

[41] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 452–461.

[42] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*. ACM, 811–820.

[43] Guy Shani, David Heckerman, and Ronen I. Brafman. 2005. An MDP-based recommender system. *Journal of Machine Learning Research* 6, (Sep. 2005), 1265–1295.

[44] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 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.

[45] Jiaxi Tang and Ke Wang. 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.

[46] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer, 776–794.

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[48] Chenyang Wang, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2020. Toward dynamic user intention: Temporal evolutionary effects of item relations in sequential recommendation. *ACM Transactions on Information Systems (TOIS)* 39, 2 (2020), 1–33.

[49] Chenyang Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. 2020. Make it a chorus: Knowledge- and time-aware item modeling for sequential recommendation. In *Proceedings of the 43trd International ACM SIGIR Conference*. ACM.

[50] Chenyang Wang, Weizhi Ma, Min Zhang, Chuancheng Lv, Fengyuan Wan, Huijie Lin, Taoran Tang, Yiqun Liu, and Shaoping Ma. 2021. Temporal cross-effects in knowledge tracing. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 517–525.

[51] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2019. Modeling item-specific temporal dynamics of repeat consumption for recommender systems. In *The World Wide Web Conference*. ACM, 1977–1987.

[52] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet Orgun. 2019. Sequential recommender systems: Challenges, progress and prospects. In *28th International Joint Conference on Artificial Intelligence, IJCAI 2019*. International Joint Conferences on Artificial Intelligence, 6332–6338.

[53] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 165–174.

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

[55] Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. CLEAR: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466* (2020).

[56] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3733–3742.

[57] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2021. Self-supervised hypergraph convolutional networks for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4503–4511.

[58] Xu Xie, Fei Sun, Zhaoyang Liu, Jinyang Gao, Bolin Ding, and Bin Cui. 2020. Contrastive pre-training for sequential recommendation. *arXiv e-prints* (2020), arXiv–2010.

[59] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*. 2048–2057.

[60] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H. Chi, Steve Tjoa, Jieqi Kang, et al. 2021. Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4321–4330.

[61] Lu Yu, Chuxu Zhang, Shangsong Liang, and Xiangliang Zhang. 2019. Multi-order attentive ranking model for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5709–5716.

[62] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1469–1478.

[63] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. 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.

[64] Chang Zhou, Jianxin Ma, Jianwei Zhang, Jingren Zhou, and Hongxia Yang. 2021. Contrastive learning for debiased candidate generation in large-scale recommender systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3985–3995.

[65] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 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.

Received 1 June 2021; revised 5 January 2022; accepted 24 February 2022