Relative Contrastive Learning for Sequential Recommendation with Similarity-based Positive Pair Selection

Zhikai Wang
Shanghai Jiao Tong University
Shanghai, China
Cloudcatcher.888@sjtu.edu.cn

Yanyan Shen
Shanghai Jiao Tong University
Shanghai, China
shenyy@sjtu.edu.cn

Zexi Zhang
Shanghai Jiao Tong University
Shanghai, China
zhang-zexi@sjtu.edu.cn

Li He
Meituan
Shanghai, China
heli18@meituan.com

Yichun Li
Meituan
Shanghai, China
yichun.li@meituan.com

Hao Gu
Meituan
Shanghai, China
guhao02@meituan.com

Yinghua Zhang
Meituan
Shanghai, China
yzhangdx@outlook.com

ABSTRACT

Contrastive Learning (CL) enhances the training of sequential recommendation (SR) models through informative self-supervision signals. Existing methods often rely on data augmentation strategies to create positive samples and promote representation invariance. Some strategies such as item reordering and item substitution may inadvertently alter user intent. Supervised Contrastive Learning (SCL) based methods find an alternative to augmentation-based CL methods by selecting same-target sequences (interaction sequences with the same target item) to form positive samples. However, SCL-based methods suffer from the scarcity of same-target sequences and consequently lack enough signals for contrastive learning. In this work, we propose to use similar sequences (with different target items) as additional positive samples and introduce a **R**elative **C**ontrastive **L**earning (RCL) framework for sequential recommendation. RCL comprises a dual-tiered positive sample selection module and a relative contrastive learning module. The former module selects same-target sequences as strong positive samples and selects similar sequences as weak positive samples. The latter module employs a weighted relative contrastive loss, ensuring that each sequence is represented closer to its strong positive samples than its weak positive samples. We apply RCL on two mainstream deep learning-based SR models, and our empirical results reveal that RCL can achieve 4.88% improvement averagely than the state-of-the-art SR methods on five public datasets and one private dataset. The code can be found at https://github.com/Cloudcatcher888/RCL.

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679681

CCS CONCEPTS

• **Computing methodologies** → **Knowledge representation and reasoning**; • **Information systems** → **Social recommendation**.

KEYWORDS

Contrastive learning, Self-supervised learning, Sequential recommendation

ACM Reference Format:
Zhikai Wang, Yanyan Shen, Zexi Zhang, Li He, Yichun Li, Hao Gu, and Yinghua Zhang. 2024. Relative Contrastive Learning for Sequential Recommendation with Similarity-based Positive Pair Selection. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3627673.3679681

1 INTRODUCTION

Sequential Recommendation models [8, 15–17, 21, 21, 24, 24, 25, 25–27, 29, 30, 38] predict a user's subsequent interaction based on their historical sequence, which aims at capturing both short-term preferences and long-term evolving interests. However, sequential recommendation models like GRU4Rec [12] and SASRec [15] encounter challenges of the inherent sparsity and noise within the data. In response, Contrastive Learning (CL) based methods [5, 18, 20, 35], exemplified by recent advancements [2, 3, 13], leverage diverse views to enhance the learned representations of sequences, offering a potential solution to address these data-related limitations in sequential recommendation models.

While CL aims to enhance sequence representations by maximizing agreement among augmented views of the same sequence and distancing views of different sequences, current CL-based methods [4, 6, 7, 10, 22, 34, 35] often rely on instinctual identification of random augmentation operations, like random sequence or model perturbations ('crop' and 'mask' operations). The process of identifying effective augmentation operations requires specific domain knowledge and meticulous design, potentially causing interference

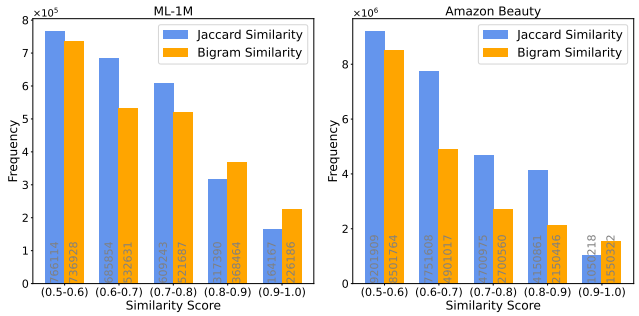


Figure 1: Similarity frequency histograms of sequence pairs with different target items on ML-1M and Amazon Beauty datasets.

with the original user intent and introducing additional noise during model training [9, 23].

Recently, supervised contrastive learning (SCL) based methods [14, 23, 28] suggest choosing same-target sequences (interaction sequences with the same target item) as positive samples. They leverage real sequences instead of augmented sequences and aim at improving the alignment among sequences sharing the same intents. For instance, ContraRec [28] introduces a contrastive learning task to encourage same-target sequences to have similar representations, called context-context contrast. In a similar vein, DuoRec [23] argues same-target sequences natually contain the same intents and designs a supervised contrastive loss to draw the representations of same-target sequences closer. However, these approaches encounter limitations due to the scarcity of same-target sequences. In the ML-1M and Amazon Beauty datasets, 47.5% and 52.7% of the sequences do not have any same-target sequences, respectively. Consequently, SCL based methods cannot obtain supervised contrast signals on these sequences with no same-target sequences, leading to limited performance improvement.

In this paper, we propose to treat similar sequences (with different target items) as additional positive samples, which has two reasons. First, a certarin proportion of sequences with different target items actually exhibit a considerable degree of similarity, the representations of which should be naturally close in the latent feature space. We provide the similarity frequency histograms of sequence pairs with different target items of ML-1M and Amazon Beauty in Figure 1. We use the Jaccard and Bigram Similarity [1] (two adjacent items are used as one gram) as the sequence similarity metrics. We find top 5.31%/4.28% of sequence pairs have Jaccard Similarity larger than 0.7 on ML-1M/Amazon Beauty, each pair of which can be regarded as similar sequences and naturally should have close representations. Second, similar sequences can serve as a supplement to the positive samples with the same target item. In ML-1M and Amazon Beauty, 100% of the sequences have at least 319/744 similar sequences (Jaccard similarity is larger than 0.7), ensuring that all sequences have corresponding positive samples. Though treating similar sequences as additional positive samples is beneficial, it is challenging to choose suitable similarity metrics and an effective sampling strategy to select similar sequences that truly share the same intents like same-target sequences.

It is further required to consider how to use both same-target sequences and similar sequences as positive samples. Intuitively, the target item directly reflects a user's future intent, while an interaction sequence represents the user's past intent, which serves as the indirect reflection of his/her future intent. Consequently, we give precedence to same-target sequences as strong positive samples, while treating similar sequences as weak positive samples. However, it is also challenging to design an appropriate contrastive learning loss function to ensures that each sequence is represented closer to its strong positive samples than its weak positive samples. If we directly use an infoNCE loss and treat strong/weak positive samples as the numerator/denominator of the infoNCE loss, the loss will inevitably push the representations of weak positive samples too far away from the center sequence.

To tackle the two challenges mentioned above, we introduce a **R**elative **C**ontrastive **L**earning (RCL) method for sequential recommendation that incorporates a dual-tiered positive sample selection module and a relative contrastive learning module. Specifically, the positive sample selection module resolves the first challenge, which introduces a weighted sequence sampling strategy based on different order sensitive/insensitive similarity metrics to select similar sequences truly sharing the same intents. In conjunction with this, the relative contrastive learning module utilizes a weighted relative contrastive loss to overcome the second challenge, which adds a loss boundary on general infoNCE loss to effectively control the relative loss magnitudes of strong positive samples and weak positive samples. Extensive experiments on six real datasets demonstrate the effectiveness of the proposed RCL framework in terms of the recommendation performance.

The main contributions of this paper are summarized as follows.

- We present a Relative Contrastive Learning (RCL) based method for sequential recommendation, which uses both same-target sequences and similar sequences as positive samples for contrative learning to further improve the alignment among sequences sharing the same intents than SCL based methods.
- A dual-tiered positive sample selection module is proposed, which introduces a weighted sampling strategy based on several similarity metrics to select similar sequences truly sharing the same intents.
- A relative contrastive learning module is proposed to control the relative loss magnitudes of strong positive samples and weak positive samples, ensuring that each sequence is represented closer to its strong positive samples than its weak positive samples.
- We validate our RCL method via comprehensive experimentation on two sequential recommendation models, i.e., SASRec and FMLP [37]. The empirical outcomes show that our approach achieves 4.88% improvement averagely against state-of-the-art methods across five public and one private datasets.

2 PRELIMINARY

2.1 Problem Definition

Sequential Recommendation (SR) aims to suggest the next item that a user is likely to interact with, leveraging their historical interaction data. Assuming that user sets and item sets are $\mathcal{U}$ and $\mathcal{V}$ respectively, user $u \in \mathcal{U}$ has a sequence of interacted items $S_u = \{v_{1,u}, \ldots, v_{|S_u|,u}\}$. $v_{i,u} \in \mathcal{V}$ $(1 \leq i \leq |S_u|)$ represents an interacted

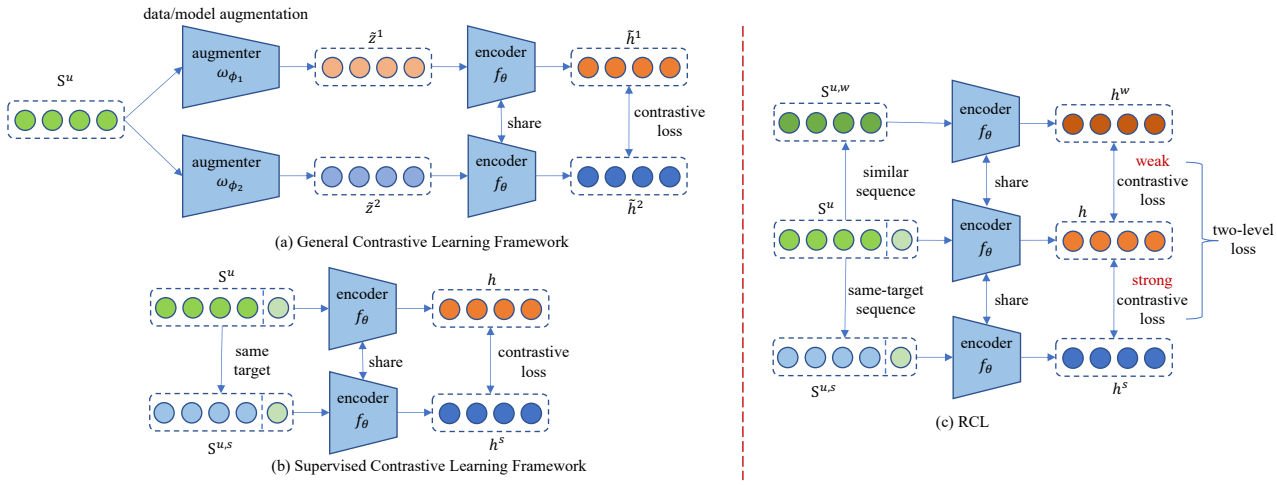


Figure 2: In the General Contrastive Learning Framework (a), the typical components include a data or model based augmentation module, a user representation encoder, and a contrastive loss function. In the Supervised Contrastive Learning Framework (b), the augmentation module is substituted with a randomly sampled same-target positive. The proposed RCL (c) differs by employing a dual-tiered positive pair selection module, which treats same-target sequences as strong positive samples and treats similar sequences as weak positive samples. A relative contrastive learning module is employed to manage the dual-tiered positive samples.

item at position i of user u within the sequence, where $|S_u|$ denotes the sequence length. Given the historical interactions S_u, the goal of SR is to recommend an item from the set of items $\mathcal{V}$ that the user u may interact with at step $|S_u|+1$:

$$\arg\max_{v' \in \mathcal{V}} P\left(v_{|S_u|+1,u} = v' \mid S_u\right). \tag{1}$$

2.2 Sequential Recommendation Model

Our method incorporates a backbone SR model comprising three key components: (1) an embedding layer, (2) a representation learning layer, and (3) a next item prediction layer.

2.2.1 Embedding Layer. Initially, the entire item set $\mathcal{V}$ is embedded into a shared space, resulting in the creation of the item embedding matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{V}| \times d}$. Given an input sequence S_u, the sequence's embedding $\mathbf{E}_u \in \mathbb{R}^{|S_u| \times d}$ is initialized, and $\mathbf{E}_u$ is defined as $\mathbf{E}_u = \left\{\mathbf{m}_1 \oplus \mathbf{p}_1, \mathbf{m}_2 \oplus \mathbf{p}_2, \ldots, \mathbf{m}_{|S_u|} \oplus \mathbf{p}_{|S_u|}\right\}$. Here, $\mathbf{m}_i \in \mathbb{R}^d$ represents the embedding of the item at position i in the sequence, $\mathbf{p}_i \in \mathbb{R}^d$ signifies the positional embedding within the sequence, $\oplus$ denotes the element-wise addition, and n denotes the sequence's length.

2.2.2 Representation Learning Layer. Given the sequence embedding $\mathbf{E}_u$, a deep neural encoder denoted as $f_\theta(\cdot)$ is utilized to learn the representation of the sequence. The output representation is calculated as:

$$\mathbf{h}_u = f_\theta\left(\mathbf{E}_u\right) \in \mathbb{R}^d. \tag{2}$$

Finally, the predicted interaction probability of each item can be calculated as:

$$\hat{\mathbf{y}} = \text{softmax}\left(\mathbf{h}_u \mathbf{M}^\top\right) \in \mathbb{R}^{|\mathcal{V}|}. \tag{3}$$

The training loss used for optimizing the sequential recommendation model is as follows:

$$\mathcal{L}^{rec} = -\sum_{u \in \mathcal{U}} \log \hat{\mathbf{y}}[v_{|S_u|+1,u}], \tag{4}$$

where $v_{|S_u|+1,u}$ denotes the ground-truth target item of user u.

3 METHODOLOGY

As depicted in Figure 2(a), the general contrastive learning framework typically comprises a stochastic augmentation module, a user representation encoder, and a contrastive loss function. The supervised contrastive learning framework, as illustrated in Figure 2(b), discards the stochastic augmentation module and chooses same-target sequences as positive samples. The proposed RCL, as shown in Figure 2(c), employs same-target sequences as strong positive samples and similar sequences as weak positive samples for each sequence, respectively. RCL implements a dual-tiered positive sample selection module, effectively capturing inherent similarities in user intent across sequences. Furthermore, RCL introduces a weighted relative contrastive learning module to ensure that each sequence is represented closer to its strong positive samples than its weak positive samples.

3.1 Dual-tiered Positive Sample Selection

In this section, we delve into RCL's method for selecting positive samples. The SCL-based methods [14, 23] have cautioned against the use of data or model-based augmentation in positive sample selection, which could potentially disrupt the inherent user intents. Instead, they try to cluster sequences with identical intents by using the same target sequences as positive samples in contrastive learning paradigm. However, these methods encounter limitations due to the scarcity of same-target sequences. We observe that some

negative samples sampled by SCL-based method [23] in each mini-batch actually have high Jaccard or Bigram similarity. Intuitively, the representations of the sequences with high similarity should be close in the latent feature space. Treating the highly similar sequences as negative samples may incorrectly push the representations away and hence degrade the learned representations. Instead, these similar sequences can be used as positive samples to enlarge the original positive sample set.

Furthermore, considering the interaction sequence as a representation of a user's historical preferences and the target item as an indicator of their future interaction intent, we prioritize treating same-target sequences as strong positive samples ($\mathcal{SP}_u$) and similar sequences as weak positive samples ($\mathcal{WP}_u$) for each sequence S_u. In what below, we will discuss how we select strong positive samples and weak positive samples.

3.1.1 Same-target Strong Positive Sample Selection. In this section, we will introduce how RCL selects the same-target sequences as strong positive samples briefly. For each sequence S_u, its strong positive samples can be formally defined as:

$$\mathcal{SP}_u = \{S_{u,a} | v_{|S_{u,a}|+1,a} = v_{|S_u|+1,u}\}, \tag{5}$$

where $S_{u,a}$ is the sequence sharing the same target item with S_u. Following the SCL-based methods [14, 23], RCL randomly picks one sequence $S_{u,a}$ from $\mathcal{SP}_u$ per iteration if $\mathcal{SP}_u$ is not empty.

3.1.2 Similarity-based Weak Positive Pair Selection. In this section, we will discuss how RCL selects similar sequences as weak positive samples $\mathcal{WP}_u$ for sequence S_u. We first introduce several similarity metrics to assess the relationship between any two user interaction sequences, S_{u_1} and S_{u_2}.

- *Jaccard Similarity:* The Jaccard Similarity between two interaction sequences can be calculated as follows:

$$s_{u_1,u_2} = \text{Jaccard Similarity}(A_{u_1}, A_{u_2}) = \frac{|A_{u_1} \cap A_{u_2}|}{|A_{u_1} \cup A_{u_2}|}, \tag{6}$$

  where A_{u_1} represents the set of unique elements in the first interaction sequence S_{u_1}. A_{u_2} represents the set of unique elements in the second interaction sequence S_{u_2}. The Jaccard Similarity measures the proportion of common elements between the sequences relative to the total unique elements in both sequences.
- *TF-IDF:* Drawing inspiration from text analysis, we conceptualize interaction sequences as sentences and items as individual words. Then the weight for each item v in sequence S_u is $w_{v,u} = tf_{v,u} \log\left(\frac{N}{df_v}\right)$, where $tf_{v,u}$ is the frequency of v in u and df_v is the frequency of v over the whole dataset. Each sequence can be transformed as item weight vector $\mathbf{w}_u = [w_{1,u}, \cdots, w_{|\mathcal{V}|,u}]$. We calculate the cosine similarity between each two vectors $\mathbf{w}_{u_1}$ and $\mathbf{w}_{u_2}$ as the sequence similarity:

$$s_{u_1,u_2} = \frac{\mathbf{w}_{u_1}^\top \mathbf{w}_{u_2}}{|\mathbf{w}_{u_1}||\mathbf{w}_{u_2}|}. \tag{7}$$

- *N-grams Similarity:* Compared to Jaccard Similarity, it considers the shared subsequences (N-grams) of a specific length (n). Formally, the N-grams similarity can be calculated as follows:

$$s_{u_1,u_2} = \frac{|A_{u_1,ngram} \cap A_{u_2,ngram}|}{|A_{u_1,ngram} \cup A_{u_2,ngram}|}, \tag{8}$$

  where $A_{u_1,ngram}$ and $A_{u_2,ngram}$ represent the sets of unique N-grams in interaction sequence S_{u_1} and S_{u_2}, respectively.
- *Levenshtein Distance:* This similarity metric refers to the minimum number of insertions, deletions, and substitutions required to transform one sequence into the other, which measures the dissimilarity between the sequences S_{u_1} and S_{u_2}, where a smaller distance indicates greater similarity:

$$d(S_{u_1}, S_{u_2}) = \min \begin{cases} d(S'_{u_1}, S'_{u_2}) + c(S_{u_1}[m], S_{u_2}[n]) \\ d(S_{u_1}, S'_{u_2}) + 1 \\ d(S'_{u_1}, S_{u_2}) + 1, \end{cases}$$

  where $S'_{u_{1/2}}$ are the string $S_{u_{1/2}}$ with their last characters removed. m and n are the lengths of $S_{u_{1/2}}$. $c(x, y)$ is a function that returns 0 if characters x and y are the same, and 1 otherwise.
- *Semantic Distance:* This similarity metric refers to the cosine similarity of sequence representations:

$$s_{u_1,u_2} = \frac{\mathbf{h}_{u_1}^\top \mathbf{h}_{u_2}}{|\mathbf{h}_{u_1}||\mathbf{h}_{u_2}|}, \tag{9}$$

  where $\mathbf{h}_{u_1}$ and $\mathbf{h}_{u_2}$ denote the representations of the interaction sequences S_{u_1} and S_{u_2}.

Jaccard Similarity and *TF-IDF* both disregard order, focusing solely on the count of interaction items. *N-grams Similarity*, on the other hand, is sensitive to local order, with larger N values increasing this order sensitivity. Meanwhile, *Levenshtein Distance* and *Semantic Distance* are explicitly sensitive to the order of elements within the sequences. The similarity calculation methods are not limited to the aforementioned five types. Our primary innovation is not the design of similarity metrics but rather in effectively leveraging weak positives for contrastive learning. We believe that the performance on the basic similarity metrics can better validate the advantages of the overall method. The choice of similarity metrics is examined in the ablation study (Section 4.3).

Then we aim at selecting weak positive samples for each sequence S_u using one similarity metric provided above. A direct strategy is choosing the sequence $S_{u,b}$ with the highest similarity score as the weak positive sample. However, the selected positive samples remain fixed across different epochs and other sequences with high similarity are not utilized for training. To refine this strategy, for each interaction sequence S_u, we now select all sequences with the top α highest similarity scores as weak positive samples $\mathcal{WP}_u$. Here, α is a hyper-parameter. We randomly choose one sequence $S_{u,b}$ from $\mathcal{WP}_u$ in each iteration during training. Intuitively, a sequence with higher similarity is preferred to be chosen. Consequently, the sampling probability for each sequence $S_{u,b}$ in $\mathcal{WP}_u$ is set to be proportional to the similarity score, which is formally defined as:

$$p_b = \frac{s_{u,b}}{\sum_{c \in \mathcal{WP}_u} s_{u,c}}, \tag{10}$$

where $s_{u,b}$ is the similarity score between sequences S_u and $S_{u,b}$, and the denominator aggregates the similarity scores of all positives in the weak positive set of sequence S_u.

3.2 Relative Contrastive Learning

In this section, we propose a relative contrastive learning module aimed at ensuring that each sequence is represented closer to its

strong positive samples than its weak positive samples. Generally, for a sequence S_u (referred to as center sequence), we aim to satisfy the condition for $S_{u,a} \in \mathcal{SP}_u$ and $S_{u,b} \in \mathcal{WP}_u$:

$$\mathbf{h}_u^\top \mathbf{h}_a > \mathbf{h}_u^\top \mathbf{h}_b, \tag{11}$$

where $\mathbf{h}_u$,$\mathbf{h}_a$, and $\mathbf{h}_b$ are the representations of sequences S_u, $S_{u,a}$, and $S_{u,b}$, respectively.

We propose to use an infoNCE-based loss function [36] for contrastive learning, which is originally designed to control the relative magnitudes of losses across different hierarchical levels of labels. Although in our scenario the labels do not overlap among different sequences, the concept of regulating the relative sizes of losses is still applicable. We therefore categorize strong positive samples as high-level labels and weak positive samples as low-level labels. To ensure that the similarity score $\mathbf{h}_u^\top \mathbf{h}_a$ is greater than the score $\mathbf{h}_u^\top \mathbf{h}_b$, we implement a constraint on the loss associated with weak positive samples. Specifically, we define the loss between S_u and $S_{u,a}$ as:

$$\mathcal{L}^{\text{pair}}(S_u, S_{u,a}) = -\log \frac{\exp(\mathbf{h}_u \cdot \mathbf{h}_a/\tau)}{\sum_{S_c \in A \setminus \{S_u\}} \exp(\mathbf{h}_u \cdot \mathbf{h}_c/\tau)}, \tag{12}$$

where A denotes all sequences in a batch and τ denotes the temperature parameter. The maximum loss (the largest distance) within strong positive samples $\mathcal{SP}_u$ is denoted as:

$$\mathcal{L}_u^{max} = \max_{S_{u,a} \in \mathcal{SP}_u} L^{\text{pair}}(S_u, S_{u,a}). \tag{13}$$

Consequently, the total loss for both strong positive samples and weak positive samples becomes:

$$\begin{aligned} \mathcal{L}^{RCL} = &\sum_{u \in \mathcal{U}} \mathcal{L}^{\text{pair}}(S_u, S_{u,a}) \\ &+ \sum_{u \in \mathcal{U}} \max\left(\mathcal{L}^{\text{pair}}(S_u, S_{u,b}), \mathcal{L}_u^{max}\right). \end{aligned} \tag{14}$$

The second term ensures that if the loss of any weak positive pair $L^{\text{pair}}(S_u, S_{u,b})$ is smaller than the largest loss among strong positive samples L_u^{max} (the boundary), the gradient will pass through L_u^{max} and push the boundary closer to sequence S_u. By controlling the upper bound of $\mathcal{L}^{\text{pair}}(S_u, S_{u,b})$, we ensure that all pairwise distances are collectively pushed towards the center, allowing better control over the contrastive objective. If the set of strong positive samples $\mathcal{SP}_u$ is empty, L^{RCL} will solely consider weak positive samples and is simplified to:

$$\mathcal{L}^{RCL} = \sum_{u \in \mathcal{U}} \mathcal{L}^{\text{pair}}(S_u, S_{u,b}). \tag{15}$$

In SCL methods [23, 28], augmentation is still necessary in case there are no positive samples with the same target item. However, in our framework, this scenario can be prevented.

3.2.1 Similarity-Based Re-weighting. Intuitively, higher similarity between sequences correspond to higher confidence in contrastive learning, thus necessitating a higher weight in the loss function for both positive samples and negative samples. To achieve this, we modify the loss between S_u and $S_{u,a}$ into a weighted version:

$$\mathcal{L}^{\text{weighted}}(S_u, S_{u,a}) = -\log \frac{s_{u,a} \exp(\mathbf{h}_u \cdot \mathbf{h}_a/\tau)}{\sum_{S_c \in A \setminus \{S_u\}} s_{u,c} \exp(\mathbf{h}_u \cdot \mathbf{h}_c/\tau)}, \tag{16}$$

where $s_{u,a}$ denotes the similarity score between S_u and $S_{u,a}$. Thus L^{RCL} will be updated as:

$$\begin{aligned} \mathcal{L}^{wRCL} = &\sum_{u \in \mathcal{U}} \mathcal{L}^{\text{pair}}(S_u, S_{u,a}) \\ &+ \sum_{u \in \mathcal{U}} \max\left(\mathcal{L}^{\text{weighted}}(S_u, S_{u,b}), \mathcal{L}_u^{max}\right). \end{aligned} \tag{17}$$

The overall objective of the joint learning is defined as:

$$\mathcal{L}^{total} = \mathcal{L}^{rec} + \lambda \mathcal{L}^{wRCL}, \tag{18}$$

where λ is a hyper-parameter used to control the magnitude of the relative contrastive loss. We use $\mathcal{L}^{wRCL}$ as the default in performance comparison. The whole training algorithm is in Algorithm 1.

Algorithm 1: The RCL Algorithm

Input: Training dataset $\{S_u\}_{u=1}^{|\mathcal{U}|}$, hyper-parameters λ,α
Output: Recommendation lists

```
Calculate the similarities between any two sequences in {S_u}_{u=1}^{|U|};
Collect strong positive samples SP_u and weak positive samples WP_u for each sequence S_u[1];
for each minibatch do
    Calculate the recommendation loss by Eq. (4);
    Calculate the RCL loss by Eq. (17);
    Calculate the joint loss by Eq. (18);
    Jointly optimize the overall objective;
end
```

3.3 Complexity Analysis

RCL introduces no additional parameters beyond generalized contrastive learning in Figure 2(a), which involves $|\mathcal{V}| \times d + |\theta|$ parameters. d and θ denote the embedding size and parameters of the sequence representation encoder. Given the total sequence count N, the sequence similarity calculation can be performed in parallel using a GPU, where each thread of the GPU is responsible for calculating one sequence's similarities with all other sequences. The GPU memory only needs to store αN^2 similarities (along with their corresponding sequence IDs), which will be replaced by newly calculated similarities with larger values. The sequence similarity calculation has a time complexity of $O\left(N^2 \times \frac{T(s)}{R}\right)$, where $T(s)$ refers to the time cost for one sequence pair. R is the number of threads in the GPU, which is a large value. For *Semantic Distance*, $T(s)$ is d^2, where d is the embedding and hidden vector dimension. For the other four metrics, $T(s)$ is $\overline{|S|}$, where $\overline{|S|}$ is the average sequence length. The GPU memory cost of sequence similarity calculation is αN^2, where α is the similarity threshold and is set to a small value. In practical recommender systems, narrowing down the scope to calculate the sequence similarity within a specific city or day can further decrease the complexity.

[1]If Semantic Distance is selected as the similarity metric, the sequence similarity becomes dynamically adaptive, requiring recalculation in each iteration to capture its evolving nature.

Table 1: Statistics of the datasets after preprocessing.

	# Users	# Items	# Avg. Length	# Actions	Sparsity
Beauty	22,363	12,101	8.9	198,502	99.93%
Sports	35,598	18,357	8.3	296,337	99.95%
ML-1M	6,041	3,417	165.5	999,611	95.16%
ML-20M	138,493	27,278	144.4	20,000,263	99.47%
Yelp	30,499	20,068	10.4	317,182	99.95%
Life Service	2,508,449	276,331	40.8	102,399,201	99.99%

4 EXPERIMENTS

In experiments, we will answer the following research questions:

- **RQ1** How does the RCL perform compared with the state-of-the-art methods?
- **RQ2** How does each component of the RCL contribute to its effectiveness?
- **RQ3** How do hyperparameters influence the performance of RCL?
- **RQ4** Where do the improvements of the RCL come from?
- **RQ5** How does RCL perform on the online sequential recommendation platform?

4.1 Setup

4.1.1 Datasets. The experiments cover six benchmark datasets, detailed in Table 1 after preprocessing:

- **Amazon Beauty** and **Sports**[2] utilize the widely-used Amazon dataset with two sub-categories as previous baselines.
- **MovieLens-1M/20M** (ML-1M/20M)[3] are two versions of a popular movie recommendation dataset with different sizes.
- **Yelp**[4] is widely used for business recommendation. Similar to previous works [20, 23], the interaction records after Jan. 1st, 2019 are used in our experiments.
- **Life Service** is a private sequential recommendation dataset, which is collected from Meituan Dianping platform for local life services such as restaurants and entertainment places. We focus on user interactions within a single day in a specific city.

We follow the settings of previous works [14, 20, 23]. All interactions are treated as implicit feedback, filtering out users or items appearing fewer than five times. The maximum sequence length for the ML-1M/20M datasets is set to 200, whereas for the other four datasets, the maximum sequence length is set to 50.

For evaluation, we use Top-K Hit Ratio (HR@K) and Top-K Normalized Discounted Cumulative Gain (NDCG@K) across K values of $\{5, 10\}$, evaluating rankings across the entire item set for fair comparison, following established methodologies [23].

4.1.2 Baselines. We compare our proposed RCL with various existing baselines. We consider three categories (I: base model without CL, II: augmentation-based method, III: SCL-based method) of comparison methods as follows.

- **SASRec (I)** [15] is a single-directional self-attention model. It is a strong baseline in the sequential recommendation.
- **CL4SRec (II)** [35] uses item cropping, masking, and reordering as augmentations for contrastive learning. It is the first contrastive learning method for sequential recommendation.
- **CT4Rec (II)** [4] offers a model augmentation-based contrastive learning approach for sequential recommendation, which adds two extra training objectives that ensure consistency in user representations across different dropout masks during training.
- **MCLRec (II)** [20] is an augmentation-based method for sequential recommendation, which contrasts data/model-level augmented views for adaptively capturing the informative features hidden in stochastic data augmentation.
- **DuoRec (III)** [23] utilizes both sequences with same target item and model-level augmented sequences as positive samples for contrastive learning.
- **ContraRec (III)** [28] introduces a supervised contrastive learning method for sequential recommendation by utilizing sequences with same target item as positive samples.
- **HPM (III)** [14] is a unified framework for sequential recommendation that leverages contrastive learning to optimize hierarchical self-supervised signals in user interaction sequences, accommodating various base sequence encoders.

4.1.3 Implementation. RCL and all baselines use SASRec as the base model. The embedding size and hidden size are set to 64. The numbers of layers and heads in the Transformer are set to 2. The Dropout [23] rate on the embedding matrix and the Transformer module is chosen from $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. The training batch size is set to 256. We use the Adam [15] optimizer with the learning 0.001. λ in Equation (18) is chosen from $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. α in Section 3.1.2 is chosen from $\{0.025, 0.05, 0.1, 0.15, 0.2, 0.25\}$. Temperature τ in Equation (12) is chosen from $\{0.01, 0.05, 0.1, 0.5, 1, 5\}$.

4.2 RQ1: Overall Performance

In this study, we assess the overall performance of RCL against various baselines, with the comparative results delineated in Table 2. For this comparison, we utilize the 2-gram similarity variant of RCL. From the collected data, we make several observations. First, the augmentation-based methods—CL4SRec, MCLRec, and CT4Rec—consistently outperform base model that do not utilize contrastive learning, highlighting the significance of augmentation-based contrastive learning in enhancing sequence representations. Second, among SCL-based methods, HPM outshines other augmentation-based methods. Third, RCL achieves 4.36%, 5.57%, 3.83%, 3.88%, 4.91%, and 6.42% improvements over state-of-the-art methods across six datasets, evidencing the effectiveness of the proposed dual-tiered positive sample selection module and relative contrastive learning module. Section 3.3 introduces a way to use the GPU for similarity calculating acceleration. By setting the thread number of the GPU as 256 and the similar sequence ratio α as 0.05, GPU-based sequence similarity calculation can reduce 99.0% time cost and 95.1% memory cost on two largest public datasets, **ML-20M** and **Yelp**, in average compared to the CPU-based method.

[2] https://jmcauley.ucsd.edu/data/amazon/
[3] https://grouplens.org/datasets/movielens/1m/
[4] https://www.yelp.com/dataset

Table 2: Overall performance. Bold scores represent the highest results of all methods. Underlined scores stand for the highest results from previous methods. The RCL achieves the state-of-the-art result among all baseline methods.

Dataset	Metric	SASRec	CL4SRec	CT4Rec	MCLRec	DuoRec	ContraRec	HPM	RCL	Improv.
Beauty	HR@5	0.0365	0.0401	0.0575	0.0581	0.0546	0.0551	0.0572	**0.0601** ± 0.0011	5.07%
	HR@10	0.0627	0.0683	0.0856	0.0861	0.0845	0.0855	0.0860	**0.0898** ± 0.0005	3.10%
	NDCG@5	0.0236	0.0263	0.0342	0.0352	0.0352	0.0354	0.0361	**0.0377** ± 0.0008	4.43%
	NDCG@10	0.0281	0.0317	0.0428	0.0446	0.0443	0.0442	0.0454	**0.0476** ± 0.0015	4.84%
Sports	HR@5	0.0218	0.0227	0.0311	0.0328	0.0326	0.0328	0.0334	**0.0354** ± 0.0015	5.99%
	HR@10	0.0336	0.0374	0.0479	0.0501	0.0498	0.0502	0.0506	**0.0528** ± 0.0014	4.35%
	NDCG@5	0.0127	0.0149	0.0189	0.0204	0.0208	0.0206	0.0214	**0.0227** ± 0.0006	6.07%
	NDCG@10	0.0169	0.0194	0.0260	0.0260	0.0262	0.0264	0.0271	**0.0287** ± 0.0015	5.90%
ML-1M	HR@5	0.1087	0.1147	0.1987	0.2041	0.2038	0.2039	0.2043	**0.2113** ± 0.0017	3.43%
	HR@10	0.1904	0.1975	0.2904	0.2933	0.2946	0.2944	0.2951	**0.3045** ± 0.0021	3.19%
	NDCG@5	0.0638	0.0662	0.1346	0.1389	0.1390	0.1392	0.1402	**0.1469** ± 0.0013	4.78%
	NDCG@10	0.0910	0.0928	0.1634	0.1683	0.1680	0.1682	0.1696	**0.1763** ± 0.0010	3.95%
ML-20M	HR@5	0.1143	0.1287	0.2079	0.2102	0.2098	0.2096	0.2113	**0.2193** ± 0.0014	3.79%
	HR@10	0.2152	0.2271	0.2802	0.2995	0.3001	0.3007	0.3011	**0.3113** ± 0.0018	3.39%
	NDCG@5	0.0717	0.0891	0.1399	0.1421	0.1428	0.1433	0.1436	**0.1503** ± 0.0012	4.69%
	NDCG@10	0.1013	0.1131	0.1652	0.1726	0.1735	0.1739	0.1741	**0.1805** ± 0.0009	3.68%
Yelp	HR@5	0.0155	0.0233	0.0433	0.0454	0.0429	0.0439	0.0461	**0.0481** ± 0.0004	4.34%
	HR@10	0.0268	0.0342	0.0617	0.0647	0.0614	0.0618	0.0651	**0.0675** ± 0.0011	3.69%
	NDCG@5	0.0103	0.0122	0.0307	0.0332	0.0324	0.0333	0.0335	**0.0358** ± 0.0012	6.87%
	NDCG@10	0.0133	0.0151	0.0356	0.0394	0.0383	0.0388	0.0399	**0.0418** ± 0.0004	4.76%
Life Service	HR@5	0.0108	0.0117	0.0179	0.0198	0.0193	0.0193	0.0204	**0.0216** ± 0.0013	5.88%
	HR@10	0.0142	0.0161	0.0225	0.0249	0.0246	0.0248	0.0252	**0.0268** ± 0.0012	6.35%
	NDCG@5	0.0066	0.0077	0.0118	0.0142	0.0145	0.0146	0.0149	**0.0164** ± 0.0013	6.71%
	NDCG@10	0.0096	0.0112	0.0158	0.0187	0.0189	0.0185	0.0192	**0.0205** ± 0.0008	6.77%

Table 3: Performance improvement on the state-of-the-art SR model FMLP.

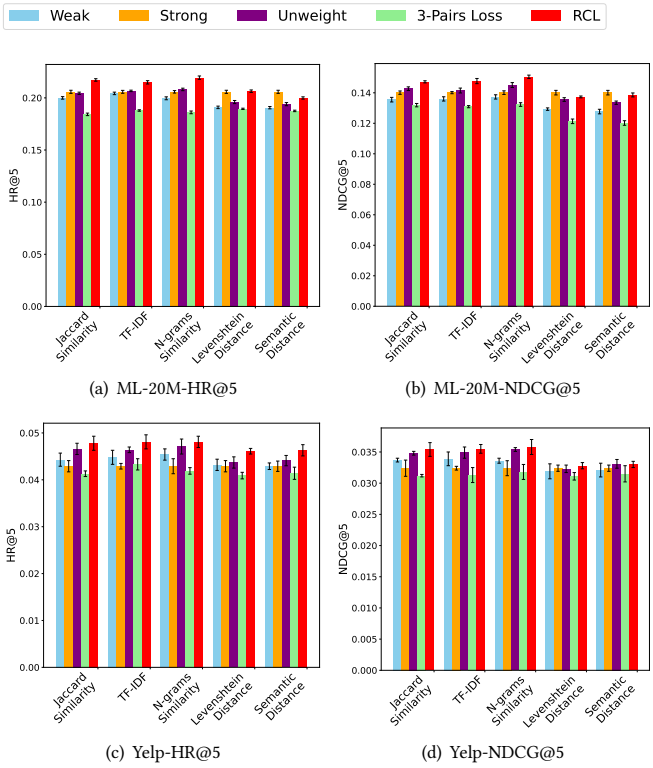
Dataset	Metric	FMLP	FMLP+HPM	FMLP+RCL
ML-20M	HR@5	0.1384	0.2154	0.2233 ± 0.0007
	HR@10	0.2398	0.3062	0.3144 ± 0.0011
	NDCG@5	0.0925	0.1477	0.1528 ± 0.0013
	NDCG@10	0.1283	0.1783	0.1826 ± 0.0007
Yelp	HR@5	0.0197	0.0473	0.0496 ± 0.0012
	HR@10	0.0321	0.0657	0.0686 ± 0.0011
	NDCG@5	0.0148	0.0357	0.0379 ± 0.0014
	NDCG@10	0.0185	0.0423	0.0452 ± 0.0008

We further conduct a supplementary performance evaluation as shown in Table 3, examining RCL's impact on a more advanced base model called FMLP. FMLP [37] is an all-MLP model utilizing a learnable filter-enhanced block for noise reduction in the embedding matrix. We report the results on the two largest public datasets, ML-20M and Yelp, and observe similar improvements on other four datasets. Table 3 shows that FMLP combined with RCL considerably improves performance by 3.97% and 5.57% on the two datasets against the state-of-the-art SCL method (HPM), respectively. These findings suggest that RCL can consistently boost the capabilities of various base models, including both transformer-based and MLP-based models.

4.3 RQ2: Ablation Study

In this section, we conduct an ablation study on two largest public datasets, **ML-20M** and **Yelp**, to assess the impact of different similarity metrics and loss functions in RCL on recommendation performance. Similar observations are made across the other four datasets and HR/NDCG@10. For similarity metrics (mentioned in Section 3.1.2), five metrics are evaluated. For loss functions, the following variants are considered (from Section 3.2):

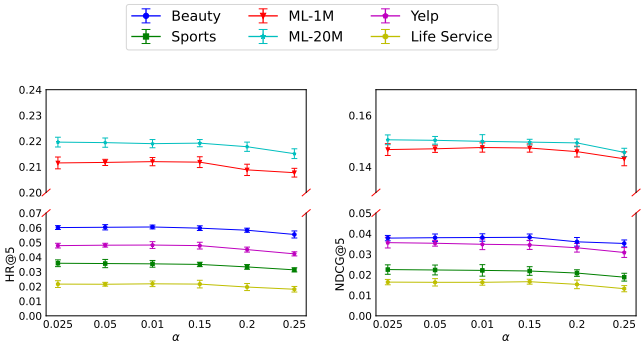
- **Weak** utilizes only weak positive samples using $\mathcal{L}^{RCL}$ from Eq. (15).
- **Strong** solely relies on strong positive samples with $\mathcal{L}^{pair}$ from Eq. (12). In cases where same-target sequences are absent, augmented user representation serves as positive samples instead [23].
- **Unweight** represents the unweighted RCL version using $\mathcal{L}^{RCL}$ in Eq. (14).
- **RCL** incorporates the proposed weighted RCL version using $\mathcal{L}^{wRCL}$ in Eq. (17).



(a) ML-20M-HR@5 (b) ML-20M-NDCG@5

(c) Yelp-HR@5 (d) Yelp-NDCG@5

Figure 3: Ablation study of different similarity metrics and loss functions.



(a) α-HR@5 (b) α-NDCG@5

Figure 4: Performance with different top similar sequence ratios α% on Beauty and Yelp datasets.

- **3-Pairs Loss** combines three infoNCE losses:

$$\mathcal{L}^{3-Pairs} = \sum_{u \in \mathcal{U}} \left(\mathcal{L}_u^{\text{S-W}} + \mathcal{L}^{\text{pair}}(S_u, S_{u,a}) + \mathcal{L}^{\text{pair}}(S_u, S_{u,b}) \right). \tag{19}$$

For sequence S_u, $\mathcal{L}_u^{\text{S-W}}$ treats same-target/similar sequences as positive/negative samples to encourage same-target sequences to have closer representations with S_u than similar sequences. $\mathcal{L}^{\text{pair}}(S_u, S_{u,a})$/$\mathcal{L}^{\text{pair}}(S_u, S_{u,b})$ treat same-target/similar sequences as positive samples and treat the other sequences except S_u in the minibatch as negative samples.

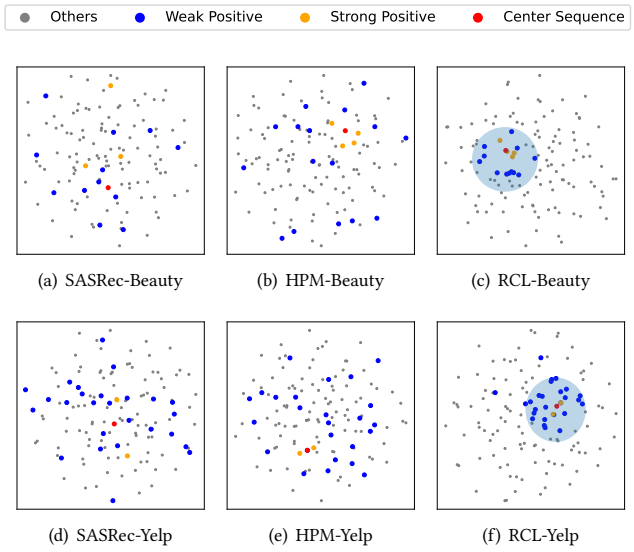
Figure 3 shows the performance of RCL using different similarity metrics and loss functions. We summarize four observations. First, both **Unweight** and **RCL** outperform **Weak** and **Strong**, indicating the effectiveness of employing both strong and weak positive samples. Using only similar sequences might introduce more noise, while relying solely on same-target sequences could introduce noise too if they're unavailable and change to augmented representation instead. Second, **RCL** performs better than **Unweight** on both datasets, highlighting the effectiveness of the similarity-based reweighting mechanism (Section 3.2). Third, **3-Pairs Loss** significantly underperforms the other four groups, probably 9 due to $\mathcal{L}^{\text{S-W}}$ in Eq. (19) pushing weak positive samples further from the center sequence S_u than negative samples. Fourth, the order-sensitive groups (Levenshtein Distance and Semantic Distance) underperform order-insensitive ones, probably because order-sensitive metrics might be overly strict by filtering sequences with many co-appearing items but differing interaction orders. However, *2-grams Similarity* slightly outperforms *Jaccard Similarity* and *TF-IDF*, suggesting the importance of considering local order.

4.4 RQ3: Parameter Sensitivity

We provide the sensitivity results of top similar sequence ratio α% on RCL performance in Figure 4. We focus on the **Beauty** and **Yelp** datasets, evaluating on HR/NDCG@5. Similar observations hold for other datasets and evaluation metrics. The top similar sequence ratio α in Section 3.1.2, chosen from $\{0.025, 0.05, 0.1, 0.15, 0.02, 0.025\}$, displays consistent performance with smaller values but exhibits roughly 10% performance drop with larger ratios. As α increases, sequences with low similarity are more likely to be sampled as weak positive samples. The performance drop might be attributed to such noisy weak positive samples.

4.5 RQ4: Case Study

In this section, we conduct a case study by acquiring the t-SNE embeddings of two center sequences along with their associated strong positive samples, weak positive samples, and negative samples for RCL. For brevity, we randomly sample 128 sequences for each center sequence. For comparison, we also provide the t-SNE embeddings of these 128 sequences in SASRec and HPM (a representative SCL method). We observe that in HPM and RCL, the strong positive samples are closest to the center sequence, indicating the infoNCE loss denoted in Eq. (12) is effective. With RCL, the weak positive samples are situated closer to the center sequence compared to negative samples but not as close as strong positive samples, which demonstrates the effectiveness of the restraint mechanism in Eq. (17). Figure 6 provides the average dot products of the center sequences with their associated strong positive samples, weak positive samples, and negative samples on six datasets respectively, further underlining the efficacy of the RCL loss function.



(a) SASRec-Beauty (b) HPM-Beauty (c) RCL-Beauty

(d) SASRec-Yelp (e) HPM-Yelp (f) RCL-Yelp

Figure 5: t-SNE results of two sequences from Beauty and Yelp for RCL and the compared baselines.

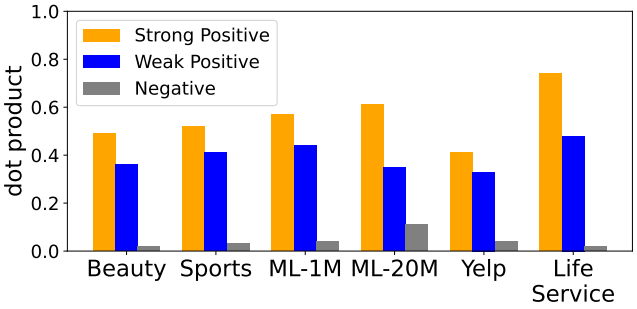


Figure 6: The average dot products of the center sequence.

Table 4: The improvements of RCL during online evaluation on the Average View Time per User (AVTU) metric.

Day	1	2	3	4	5	6	7
Improvement(%)	0.32	0.36	0.93	0.93	1.03	1.27	0.20

4.6 RQ5: A/B Testing

We further evaluate RCL on a private online recommendation platform which uses MIND [16] as the base sequential recommendation model. We use RCL for 20% users and use SCL (only strong positive samples) for the others and compare the performance between RCL and SCL. Since the online platform produces 20+ million interaction sequences every day, we only calculate similarities of sequences whose target items have the same category to guarantee the sampling efficiency. The evaluation time span is 2024/06/07-2024/06/13. This platform uses Average View Time per User (AVTU) as the main evaluation metric which indicates user engagement and the overall user experience with the platform. Table 4 shows the improvements of RCL during 7-day recommendation, which testifies the effectiveness of RCL for real recommendation systems.

5 RELATED WORK

5.1 Sequential Recommendation

Existing sequential recommendation models [8, 26, 29, 31, 33, 38] mainly rely on recurrent neural networks like GRU [12] or attention mechanisms such as transformer [6, 15, 27, 32] as sequence encoders. Some works [21, 24, 25] also incorporate graph neural networks to model sequences. Most models [8, 15] center on the next-item prediction task, which is inherently suitable for predicting subsequent items. The supervision of next-item prediction is limited to data sparsity issue [22] and auxiliary tasks such as masked prediction have a semantic gap to the recommendation task.

5.2 Contrastive Learning

Contrastive learning (CL) has found extensive applications [2, 3, 11, 13, 19] in deep learning to learn latent representations. Based on whether labels are used when generating positive samples, CL can be grouped into self-supervised CL and supervised CL. Some works [6, 7, 22, 34, 35] apply self-supervised CL for sequential recommendation. For example, CoSeRec [18] and TiCoSeRec [5] perform item-level CL based on model-based augmentations. Recently, some works[14, 23, 28] propose to apply supervised contrastive learning by taking the sequences with the same target item as positive samples and maximize their similarity. Different from existing methods, we focus on using similar sequences as supplements to positive samples and provide extra training signals for optimizing sequential recommendation models.

6 CONCLUSION

In this work, we introduce a novel framework called **R**elative **C**ontrastive **L**earning (RCL) for sequential recommendation, which treats similar sequences as additional positive samples. Our approach comprises a dual-tiered positive sample selection module, and a relative contrastive learning module. The former module utilizes same-target sequences as strong positive samples and treats similar sequences as weak positive samples. The latter module utilizes a weighted relative contrastive loss function to draw the representations of strong positive samples relatively closer to center sequence. We apply RCL to two important sequential recommendation models, and our empirical results reveal that RCL averagely achieves 4.88% improvement against the state-of-the-art methods across five public datasets and one private dataset.

ACKNOWLEDGEMENTS

Yanyan Shen is the corresponding author and is partially supported by the National Key Research and Development Program of China (2022YFE0200500) and the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102). We would like to extend our special thanks to Peng Yan and Dongbo Xi from the Dianping App Search Team for their valuable feedback on this paper. We also appreciate the strong support from the Meituan Research Collaboration Department.

REFERENCES

[1] Abdulaziz AlQatan, Leif Azzopardi, and Yashar Moshfeghi. 2020. Analyzing the Influence of Bigrams on Retrieval Bias and Effectiveness. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval* (Virtual Event, Norway). Association for Computing Machinery, New York, NY, USA, 157–160. https://doi.org/10.1145/3409256.3409831

[2] 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.

[3] Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 15750–15758.

[4] Liu Chong, Xiaoyang Liu, Rongqin Zheng, Lixin Zhang, Xiaobo Liang, Juntao Li, Lijun Wu, Min Zhang, and Leyu Lin. 2023. CT4Rec: Simple yet Effective Consistency Training for Sequential Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3901–3913.

[5] Yizhou Dang, Enneng Yang, Guibing Guo, Linying Jiang, Xingwei Wang, Xiaoxiao Xu, Qinghui Sun, and Hong Liu. 2022. Uniform Sequence Better: Time Interval Aware Data Augmentation for Sequential Recommendation. *arXiv preprint arXiv:2212.08262* (2022).

[6] Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, Philippe Cudre Mauroux, 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. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (2020), 1893–1902. https://doi.org/10.1145/3340531.3411954

[7] Chengxin Ding, Jianhui Li, Tianhang Liu, and Zhongying Zhao. 2022. Graph-Augmented Multi-Level Representation Learning for Session-based Recommendation. *2022 IEEE 8th International Conference on Cloud Computing and Intelligent Systems (CCIS)* 00 (2022), 576–580. https://doi.org/10.1109/ccis57298.2022.10016436

[8] Yike Guo, Faisal Farooq, Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. *AAAI* (2018), 1059–1068.

[9] Mohammad Al Hasan, Li Xiong, Shuqing Bian, Wayne Xin Zhao, Jinpeng Wang, and Ji-Rong Wen. 2022. A Relevant and Diverse Retrieval-enhanced Data Augmentation Framework for Sequential Recommendation. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (2022), 2923–2932. https://doi.org/10.1145/3511808.3557071

[10] Mohammad Al Hasan, Li Xiong, Jiangxia Cao, Xin Cong, Jiawei Sheng, Tingwen Liu, and Bin Wang. 2022. Contrastive Cross-Domain Sequential Recommendation. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (2022), 138–147. https://doi.org/10.1145/3511808.3557262

[11] 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.

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

[13] 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. *arXiv preprint arXiv:1808.06670* (2018).

[14] Chengkai Huang, Shoujin Wang, Xianzhi Wang, and Lina Yao. 2023. Dual Contrastive Transformer for Hierarchical Preference Modeling in Sequential Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 99–109.

[15] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[16] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-Interest Network with Dynamic Routing for Recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 2615–2623. https://doi.org/10.1145/3357384.3357814

[17] Chong Liu, Xiaoyang Liu, Rongqin Zheng, Lixin Zhang, Xiaobo Liang, Juntao Li, Lijun Wu, Min Zhang, and Leyu Lin. 2021. C^2-Rec: An Effective Consistency Constraint for Sequential Recommendation. *arXiv* (2021). https://doi.org/10.48550/arxiv.2112.06668 arXiv:2112.06668 C2 Rec.

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

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

[20] Xiuyuan Qin, Huanhuan Yuan, Pengpeng Zhao, Junhua Fang, Fuzhen Zhuang, Guanfeng Liu, Yanchi Liu, and Victor Sheng. 2023. Meta-optimized Contrastive Learning for Sequential Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 89–98. https://doi.org/10.1145/3539618.3591727

[21] Ruihong Qiu, Zi Huang, Tong Chen, and Hongzhi Yin. 2021. Exploiting positional information for session-based recommendation. *ACM Transactions on Information Systems (TOIS)* 40, 2 (2021), 1–24.

[22] Ruihong Qiu, Zi Huang, and Hongzhi Yin. 2021. Memory Augmented Multi-Instance Contrastive Predictive Coding for Sequential Recommendation. *2021 IEEE International Conference on Data Mining (ICDM)* 00 (2021), 519–528. https://doi.org/10.1109/icdm51629.2021.00063

[23] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2021. Contrastive Learning for Representation Degeneration Problem in Sequential Recommendation. *arXiv* (2021). https://doi.org/10.48550/arxiv.2110.05730 arXiv:2110.05730

[24] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the item order in session-based recommendation with graph neural networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 579–588.

[25] Ruihong Qiu, Hongzhi Yin, Zi Huang, and Tong Chen. 2020. Gag: Global attributed graph neural network for streaming session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 669–678.

[26] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. *WWW* (2010), 811–820.

[27] 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.

[28] Chenyang Wang, Weizhi Ma, Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. Sequential Recommendation with Multiple Contrast Signals. *ACM Transactions on Information Systems* 41, 1 (2023), 1–27. https://doi.org/10.1145/3522673

[29] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning Hierarchical Representation Model for NextBasket Recommendation. *SIGIR* (2015), 403–412.

[30] Zhikai Wang and Yanyan Shen. 2022. Time-aware Multi-interest Capsule Network for Sequential Recommendation. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*. SIAM, 558–566.

[31] Zhikai Wang and Yanyan Shen. 2023. Incremental Learning for Multi-Interest Sequential Recommendation. In *ICDE*. IEEE, 1071–1083.

[32] Zhikai Wang and Yanyan Shen. 2024. A Framework for Elastic Adaptation of User Multiple Intents in Sequential Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2024), 1–13. https://doi.org/10.1109/TKDE.2024.3354796

[33] Zhikai Wang, Yanyan Shen, Zibin Zhang, and Kangyi Lin. 2023. Feature Staleness Aware Incremental Learning for CTR Prediction. In *IJCAI*.

[34] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2021. Self-Supervised Hypergraph Convolutional Networks for Session-based Recommendation *(Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35)*. 4503–4511. https://doi.org/10.1609/aaai.v35i5.16578

[35] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive Learning for Sequential Recommendation. *2022 IEEE 38th International Conference on Data Engineering (ICDE)* 00 (2022), 1259–1273. https://doi.org/10.1109/icde53745.2022.00099

[36] Shu Zhang, Ran Xu, Caiming Xiong, and Chetan Ramaiah. 2022. Use all the labels: A hierarchical multi-label contrastive learning framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16660–16669.

[37] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is All You Need for Sequential Recommendation. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 2388–2399. https://doi.org/10.1145/3485447.3512111

[38] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2017. What to Do Next: Modeling User Behaviors by Time-LSTM. *IJCAI* (2017), 3602–3608.