

---

# Semantic Retrieval Augmented Contrastive Learning for Sequential Recommendation

---

Ziqiang Cui<sup>1\*</sup>, Yunpeng Weng<sup>2,3\*</sup>, Xing Tang<sup>4†</sup>, Xiaokun Zhang<sup>1</sup>, Shiwei Li<sup>2</sup>, Peiyang Liu<sup>5</sup>,

Bowei He<sup>1</sup>, Dugang Liu<sup>6</sup>, Weihong Luo<sup>3</sup>, Xiuqiang He<sup>4</sup>, Chen Ma<sup>1†</sup>,

<sup>1</sup> City University of Hong Kong <sup>2</sup> Huazhong University of Science and Technology

<sup>3</sup> Tencent <sup>4</sup> Shenzhen Technology University <sup>5</sup> Peking University <sup>6</sup> Shenzhen University

ziqiang.cui@my.cityu.edu.hk, {wengyp, lishiwei}@hust.edu.cn, xing.tang@hotmail.com,  
{dawnkun1993, dugang.ldg}@gmail.com, liupeiyang@pku.edu.cn, boweihe2-c@my.cityu.edu.hk,  
lobby66@163.com, hexiuqiang@sztu.edu.cn, chenma@cityu.edu.hk

## Abstract

Contrastive learning has shown effectiveness in improving sequential recommendation models. However, existing methods still face challenges in generating high-quality contrastive pairs: they either rely on random perturbations that corrupt user preference patterns or depend on sparse collaborative data that generates unreliable contrastive pairs. Furthermore, existing approaches typically require predefined selection rules that impose strong assumptions, limiting the model’s ability to autonomously learn optimal contrastive pairs. To address these limitations, we propose a novel approach named Semantic Retrieval Augmented Contrastive Learning (SRA-CL). SRA-CL leverages the semantic understanding and reasoning capabilities of LLMs to generate expressive embeddings that capture both user preferences and item characteristics. These semantic embeddings enable the construction of candidate pools for inter-user and intra-user contrastive learning through semantic-based retrieval. To further enhance the quality of the contrastive samples, we introduce a learnable sample synthesizer that optimizes the contrastive sample generation process during model training. SRA-CL adopts a plug-and-play design, enabling seamless integration with existing sequential recommendation architectures. Extensive experiments on four public datasets demonstrate the effectiveness and model-agnostic nature of our approach. Our code is available at <https://github.com/ziqiangcui/SRA-CL>

## 1 Introduction

Sequential recommendation aims to model user preferences based on historical behavior sequences, a task of significant value for online platforms like YouTube and Amazon. However, accurate preference modeling faces a fundamental challenge: data sparsity, as most users have only limited interaction records and most items receive little attention. To address this issue, numerous self-supervised learning techniques [39, 46, 34] have been proposed, leveraging auxiliary tasks to improve data utilization efficiency. Among these, contrastive learning has emerged as a predominant approach due to its conceptual simplicity and proven effectiveness [34, 24, 3, 22, 47]. Typically, it constructs positive sample pairs from the data and maximizes their agreement in the representation space [2].

As illustrated in Figure 1, existing contrastive learning approaches for sequential recommendation can be broadly classified into two categories: (1) inter-user contrastive learning, which contrasts sequences

---

\*Equal contribution.

†Corresponding authors.

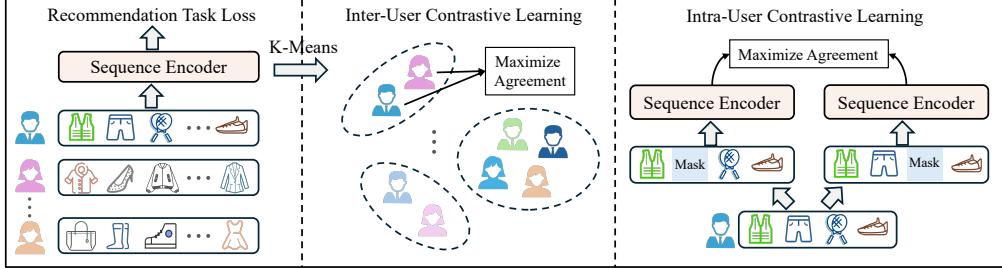


Figure 1: Illustration of existing contrastive learning methods in sequential recommendation, categorized into two main types: (1) inter-user contrastive learning and (2) intra-user contrastive learning.

from different users, and (2) intra-user contrastive learning, which contrasts different augmented views of a single user’s sequence. In the inter-user paradigm, user sequence representations are clustered using K-means, and users within the same cluster are treated as positive samples for each other [3, 17, 23]. In the intra-user paradigm, perturbations are applied to a user’s sequence to generate augmented views, and the similarity between these views is maximized [34, 21, 24, 22]. These contrastive learning methods are typically employed as auxiliary tasks alongside the primary recommendation objective and have been demonstrated to enhance recommendation performance by improving user representation learning [24, 36].

Despite their empirical success, existing methods suffer from several limitations in contrastive pair construction, which may undermine their effectiveness in recommendation scenarios. 1) **Semantic Divergence**. Many existing methods construct contrastive pairs through random augmentation operations such as random masking [34, 21] and Dropout [24]. However, in sequential recommendation where data is inherently sparse and exhibits sequential patterns, such random operations may lead to a complete change in the sequence’s semantics (i.e., user preferences). Bringing semantically different sequences closer together in embedding space may diminish the model’s ability to discriminate among distinct user preferences. Additionally, some methods determine contrastive pairs by clustering user representations derived from collaborative signals [3, 23], where users within the same cluster are considered positive pairs. However, the sparse ID signals can lead to low-quality representations and inaccurate clustering results. 2) **Unlearnability**. Existing methods rely on predefined rules to construct positive pairs, such as directly selecting users from the same cluster [3, 23, 21], or treating sequences sharing the same next item as positive pairs [24, 23]. These rigid heuristics impose strong assumptions that constrain models from autonomously learning optimal contrastive pairs. Moreover, the approach of using sequences with identical next items as positive pairs essentially replicates the recommendation objective (i.e., next-item prediction), providing no additional information gain. Therefore, the suboptimal construction of contrastive pairs in existing methods limits their effectiveness and hinders contrastive learning’s full potential.

Given these limitations, constructing high-quality contrastive samples remains a critical challenge. Semantic information, which is readily available in textual data such as product categories, brands, and descriptions, provides a promising solution. Unlike sparse behavior signals, semantic data maintains validity regardless of data volume or training dynamics, as it derives from structured knowledge rather than co-occurrence patterns [46]. Additionally, semantic features offer complementary information beyond collaborative signals. Motivated by these advantages, we propose leveraging semantic information to construct superior contrastive pairs. However, accurately capturing user preferences requires models with powerful understanding and reasoning capabilities. Recent research has shown that large language models (LLMs) can effectively understand user preferences and achieve competitive performance on sequential recommendation tasks [35]. Inspired by this, we propose to enhance contrastive learning through LLM-powered semantic retrieval.

In this paper, we propose SRA-CL (Semantic Retrieval-Augmented Contrastive Learning), a novel framework with two key innovations: 1) **Semantic-based Retrieval**. We develop a semantic-based retrieval mechanism that operates at both inter-user and intra-user levels. For inter-user contrastive learning, we leverage LLMs to process sequential user interaction histories. Each sequence is fed to the LLM in chronological order of item interactions, where each item consists of both its attributes and textual description, enabling the model to generate preference-aware semantic embeddings through comprehensive understanding of user behavior patterns. For intra-user contrastive learning, we enhance item understanding by providing LLMs with both item attributes and their contextual

sequence information, producing context-aware semantic embeddings that capture both intrinsic item properties and their relevance within the recommendation context. Subsequently, we leverage the semantic embeddings to retrieve the top- $k$  most similar users and items, constructing candidate positive sample pools for contrastive learning. 2) **Learnable Sample Synthesis**. To construct more effective contrastive samples, our framework incorporates a learnable sample synthesizer. For inter-user contrastive learning, the synthesizer dynamically generates positive samples for each user sequence by selectively combining elements from the candidate pool. This generation process is jointly optimized with the model training, ensuring the synthesized samples effectively improve representation learning.

Our main contributions are summarized as follows.

- We propose a model-agnostic framework, SRA-CL, which leverages semantic information and the capabilities of LLMs to construct better contrastive pairs, thereby improving the contrastive learning in sequential recommendation.
- We propose a semantic-based retrieval approach for contrastive pair construction that integrates dual retrieval mechanisms: user retrieval for inter-user contrastive learning and item retrieval for intra-user contrastive learning, with each mechanism maintaining its dedicated candidate pool. To further enhance this framework, we introduce a learnable sample synthesizer that optimizes the contrastive sample generation process during model training.
- We conduct extensive experiments on four public datasets to validate the superiority and model-agnostic nature of our approach, as well as to confirm the efficacy of each module.

## 2 Preliminary

### 2.1 Sequential Recommendation Task

We denote the sets of users and items by  $\mathcal{U}$  and  $\mathcal{V}$ , respectively. Each user  $u \in \mathcal{U}$  has a chronological sequence of interacted items  $\mathcal{S}_u = [v_1^u, v_2^u, \dots, v_n^u]$ , where  $v_t^u$  indicates the item that  $u$  interacted with at step  $t$ , and  $n$  is the predefined maximum sequence length. For user sequences longer than  $n$ , we retain only the most recent  $n$  items. The goal of sequential recommendation is to predict the next item  $v^+$  according to  $\mathcal{S}_u$ , which can be formulated as:

$$\arg \max_{v \in \mathcal{V}} P(v^+ = v | \mathcal{S}_u), \quad (1)$$

where the probability  $P$  represents the likelihood of item  $v$  being the next item, conditioned on  $\mathcal{S}_u$ .

### 2.2 Sequential Recommendation Backbone

Our method is model-agnostic and can be integrated with various sequential recommendation models, as demonstrated in Section 4.3. To facilitate the introduction of our approach, we adopt the transformer architecture [28] as the backbone recommendation model following previous studies [22–24].

**Embedding Layer.** We initialize an embedding matrix  $\mathbf{M} \in \mathbb{R}^{|\mathcal{V}| \times d}$  to encode item IDs, where  $|\mathcal{V}|$  represents the size of the item set and  $d$  denotes the dimensionality of the latent space. Given a user interaction sequence  $\mathcal{S}_u$ , we obtain item embeddings  $\mathbf{E}_u \in \mathbb{R}^{n \times d}$  and position embeddings  $\mathbf{P} \in \mathbb{R}^{n \times d}$ . Consequently, the input sequence  $\mathcal{S}_u$  can be represented as  $\mathbf{H}_u = \mathbf{E}_u + \mathbf{P}$ .

**Sequence Encoder.** The representation of the input sequence is then fed into  $L$  Transformer layers [28] to capture complex sequential patterns, which can be defined as follows:

$$\mathbf{H}_u^{(L)} = \text{Transformer}(\mathbf{H}_u), \quad \mathbf{h}_u = \mathbf{H}_u^{(L)}[-1]. \quad (2)$$

Here,  $\mathbf{h}_u \in \mathbb{R}^d$  represents the last position of  $\mathbf{H}_u^{(L)}$  and is selected as the final representation of  $\mathcal{S}_u$ .

**Prediction and Objective Function.** During prediction, we calculate the probability of each item using  $\hat{\mathbf{y}} = \text{softmax}(\mathbf{h}_u \mathbf{M}^T)$ , where  $\hat{\mathbf{y}} \in \mathbb{R}^{|\mathcal{V}|}$  and  $\hat{y}_v$  represents the likelihood of item  $v$  being the next item. For training, we adopt the same cross-entropy loss function as our baseline methods [22–24] to ensure fairness, where  $v^+$  denotes the ground truth item for user  $u$ .

$$\mathcal{L}_{\text{Rec}} = -\log(\hat{y}_{v^+}). \quad (3)$$

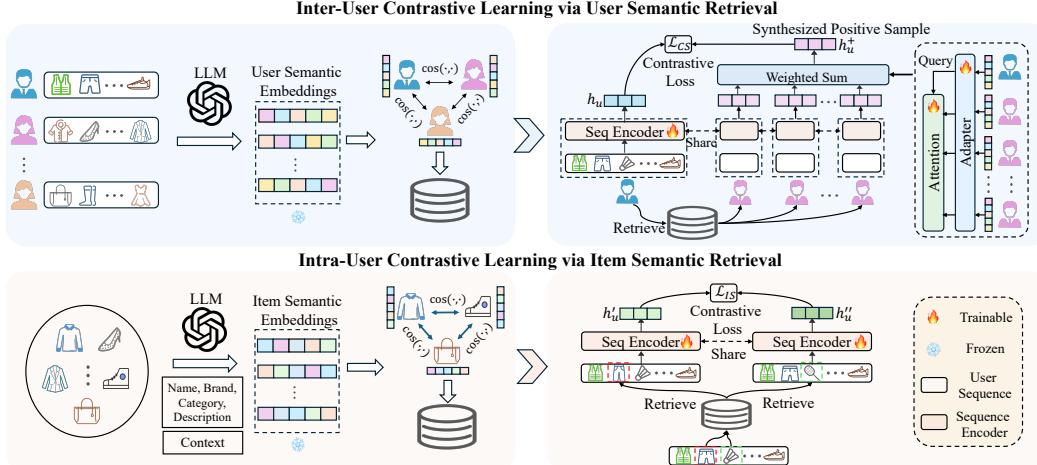


Figure 2: Overview of the proposed SRA-CL Framework.

### 3 The Framework of SRA-CL

In this section, we provide a detailed introduction to SRA-CL, which is shown in Figure 2. SRA-CL integrates inter-user contrastive learning via user semantic retrieval and intra-user contrastive learning via item semantic retrieval. To further enhance the framework, we introduce a learnable sample synthesizer that optimizes the contrastive sample generation process during model training.

#### 3.1 Inter-User Contrastive Learning via User Semantic Retrieval

SRA-CL employs semantic retrieval to generate reliable supervision signals for inter-user contrastive learning. Leveraging the advanced reasoning capabilities of LLMs, we first derive a comprehensive representation of user preferences, which are then encoded as semantic embeddings. Based on the similarity of these embeddings, we introduce a semantic-based retrieval mechanism to construct a candidate sample pool. Subsequently, a learnable contrastive sample synthesis method is employed to generate effective contrastive pairs.

**User Preference Understanding with LLMs.** Textual data (e.g., product categories, brands, and descriptions) plays a pivotal role in recommender systems by encoding rich semantic signals that reflect user preferences. Given user  $u$ 's interaction sequence  $\mathcal{S}_u$ , we extract textual attributes for each item in chronological order, preserving both content and sequential context. These features are structured into a prompt  $\mathcal{P}_u$ , where item attributes and their order explicitly guide the LLM in inferring user preferences  $\mathcal{A}_u = \text{LLM}(\mathcal{P}_u)$ . The prompt template is detailed in Figure 6.

Next, we employ a pretrained text embedding model  $\mathcal{M}$  to extract and convert the semantic information contained in the textual responses of LLMs into embeddings, which is formatted as:

$$\tilde{\mathbf{h}}_u = \mathcal{M}(\mathcal{A}_u), \quad (4)$$

where  $\tilde{\mathbf{h}}_u \in \mathbb{R}^{\tilde{d}}$  represents the semantic embedding of user preferences and  $\tilde{d}$  is the embedding size of the text embedding model  $\mathcal{M}$ . Specifically,  $\mathcal{M}$  indicates SimCSE-RoBERTa [8] in this paper due to its open-source availability and excellent sentence semantic extraction capabilities. The generated semantic embeddings are cached and remain fixed throughout the whole training process.

**Semantic-based User Retrieval.** Once the semantic embeddings of user sequences are obtained, similar users can be retrieved based on semantic similarity. For a given user sequence  $\mathcal{S}_u$ , we calculate the cosine similarity between its semantic embedding  $\tilde{\mathbf{h}}_u$  and the semantic embeddings of other users. Users are then ranked in descending order according to the computed semantic similarity. The top  $k$  users are retrieved to construct the homogeneous user pool for user  $u$ , denoted as  $\mathcal{N}_u$ .

$$\mathcal{N}_u = \{u' \in \mathcal{U} \setminus \{u\} \mid \text{rank}(\text{cosine\_similarity}(\tilde{\mathbf{h}}_u, \tilde{\mathbf{h}}_{u'})) \leq k\}, \quad (5)$$

where  $\mathcal{U} \setminus \{u\}$  denotes the set of all users except  $u$ .

**Learnable Contrastive Sample Synthesis.** Sole reliance on hard rules, such as selecting a user from the current user’s dedicated candidate pool as the positive sample, often yields suboptimal solutions (as shown in Table 2). To enhance contrastive sample construction, we introduce a learnable sample synthesizer that optimizes the contrastive sample generation process during model training. Specifically, we first map the semantic representations of user sequences through a learnable adapter. Then, in the mapped space, we employ an attention mechanism, where the current user serves as a query to compute the probability  $p_{u,u'}$  that each candidate user  $u' \in \mathcal{N}_u$  is suitable as the positive sample for the current user  $u$ . This process is formulated as:

$$w_{u,u'} = \text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W}\mathbf{h}_u \parallel \mathbf{W}\mathbf{h}_{u'}]), \quad (6)$$

$$p_{u,u'} = \text{softmax}_{u'}(w_{u,u'}) = \frac{\exp(w_{u,u'})}{\sum_{u_k \in \mathcal{N}_u} \exp(w_{u,u_k})}, \quad (7)$$

where  $\mathbf{W} \in \mathbb{R}^{d \times \tilde{d}}$  is a learnable weight matrix, and  $\parallel$  denotes the concatenation operation.  $\mathbf{a} \in \mathbb{R}^{2d}$  represents a single-layer neural network used to generate the attention score, with the LeakyReLU activation function adopted [29]. The softmax function is employed to transform the coefficients into probabilities. Based on this, we generate the composite positive contrastive sample  $\mathbf{h}_u^+$  for  $\mathbf{h}_u$  by:

$$\mathbf{h}_u^+ = \sum_{u' \in \mathcal{N}_u} p_{u,u'} \mathbf{h}_{u'}, \quad (8)$$

where  $\mathbf{h}_{u'} \in \mathbb{R}^d$  is the recommendation model’s output sequence representation for  $u'$ , as defined in Equation (2). This operation enables a fine-grained learnable selection of contrastive samples.

**Inter-User Contrastive Loss.** For each user  $u$ ,  $\mathbf{h}_u$  is the sequence representation obtained from the recommendation model. The synthetic representation  $\mathbf{h}_u^+$  is regarded as the positive sample for  $\mathbf{h}_u$ , while the remaining  $N - 1$  synthetic representations within the same batch are treated as negative samples for  $\mathbf{h}_u$ , where  $N$  is the batch size. We compute the inter-user contrastive loss  $\mathcal{L}_{\text{CS}}$  as follows:

$$\mathcal{L}_{\text{CS}} = -\log \frac{\exp(\mathbf{h}_u \cdot \mathbf{h}_u^+)}{\exp(\mathbf{h}_u \cdot \mathbf{h}_u^+) + \sum_{\mathbf{h}_u^- \in \mathbf{H}_u^-} \exp(\mathbf{h}_u \cdot \mathbf{h}_u^-)}, \quad (9)$$

where  $(\cdot)$  represents the inner product operation,  $\mathbf{H}_u^-$  denotes the set of negative samples for  $\mathbf{h}_u$ .

### 3.2 Intra-User Contrastive Learning via Item Semantic Retrieval

For intra-user contrastive learning, most existing methods apply predefined random perturbations to the original sequence to generate augmented views, which are treated as a pair of positive samples [34, 24]. A significant limitation of them is the introduction of considerable uncertainty in the semantic similarity between positive samples. This substantial variation in user sequence semantics among positive samples undermines the reliability of the contrastive learning process. To address this issue, we leverage a comprehensive understanding of both the semantic information of the item itself and the typical contexts in which the item appears. Based on this understanding, we replace certain items in the sequence with similar ones, resulting in semantic-consistent positive samples.

**Item Understanding with LLMs.** To enhance the LLM’s comprehension of items, we provide two types of input information: (1) *textual attributes of the item*, including category, brand, and description, which supply fundamental information and enable the LLM to perform a coarse-grained assessment of item similarity; and (2) *user sequences containing the given item*. By analyzing the typical contexts in which an item appears, the LLM can infer the characteristics of its potential audience. This methodology facilitates more accurate evaluations of the relationships between items in the context of sequential recommendation. Given the token limit for LLM input, we have constrained the maximum number of item-related sequences in the prompt to 10, leaving the exploration of this value for future research. Next, these two types of information for item  $v$  are integrated into a structured prompt  $\mathcal{P}_v$ , which is processed by the language model to generate the item summary  $\mathcal{A}_v = \text{LLM}(\mathcal{P}_v)$ . The detailed prompt template is illustrated in Figure 6. Then, the pretrained text embedding model  $\mathcal{M}$  is used to convert the textual responses of LLMs into embeddings:  $\tilde{\mathbf{e}}_v = \mathcal{M}(\mathcal{A}_v)$ .

**Semantic-based Item Retrieval.** Similar to user retrieval, we compute the cosine similarity between the semantic embedding of an item and those of other items. Next, Top- $k$  most semantically relevant items for item  $v$  are retrieved, which is formulated as:

$$\mathcal{N}_v = \{v' \in \mathcal{V} \setminus \{v\} \mid \text{rank}(\text{cosine\_similarity}(\tilde{\mathbf{e}}_v, \tilde{\mathbf{e}}_{v'})) \leq k\}, \quad (10)$$

**Contrastive Sample Selection.** For intra-user contrastive learning, generating two semantic-consistent augmented views of the same user sequence is crucial. Here, we employ a semantic-based item substitution approach. Specifically, for each sequence  $\mathcal{S}_u$ , we randomly select 20% of the items. For each selected item  $v$ , we substitute it with a semantically similar item sampled from its candidate pool  $\mathcal{N}_v$ . This operation yields an augmented sequence  $\mathcal{S}'_u$  derived from the original  $\mathcal{S}_u$ . By repeating this process, we obtain two augmented views, denoted as  $\mathcal{S}'_u$  and  $\mathcal{S}''_u$ , which form a positive sample pair. Critically, the substitution is not entirely random but is guided by semantic similarity, which accounts for both item attribute similarity and contextual relevance in the recommendation scenario. This reduces uncertainty and enhances semantic consistency between augmented views.

Our preliminary experiments also explored the use of learnable synthesizers (analogous to inter-contrastive learning approaches) for generating substitute items, yet yielded no measurable performance improvements (shown in Table 4). This can be attributed to the inherently higher interpretability and quantifiability of item semantics relative to user preferences. Therefore, directly identifying appropriate substitutes from semantically similar candidate pools is simpler and more reliable compared to matching users with analogous preference patterns. A more detailed analysis is provided in Appendix C.1.

**Intra-User Contrastive Loss.** For the two augmented sequences  $\mathcal{S}'_u$  and  $\mathcal{S}''_u$ , we obtain their hidden vectors  $\mathbf{h}'_u$  and  $\mathbf{h}''_u$  using the sequence encoder defined in Equation (2). Then the intra-user contrastive loss can be calculated as:

$$\mathcal{L}_{\text{IS}} = -\log \frac{\exp(\mathbf{h}'_u \cdot \mathbf{h}''_u)}{\exp(\mathbf{h}'_u \cdot \mathbf{h}''_u) + \sum_{\mathbf{h}_u^{\text{neg}} \in \mathbf{H}_u^{\text{neg}}} \exp(\mathbf{h}'_u \cdot \mathbf{h}_u^{\text{neg}})} \quad (11)$$

In a batch with a size of  $N$ , we have  $2N$  augmented sequences. Among these,  $\mathbf{h}'_u$  and  $\mathbf{h}''_u$  are positive samples of each other and are interchangeable. The remaining  $2(N - 1)$  samples excluding  $\mathbf{h}'_u$  and  $\mathbf{h}''_u$  are considered negative samples  $\mathbf{H}_u^{\text{neg}}$ .

---

**Algorithm 1** Training for SRA-CL

---

**Require:** Training data  $\{\mathcal{S}_u\}$  for all  $u \in \mathcal{U}$ ; hyperparameters  $\alpha, \beta, k$

- 1: Obtain user semantic embeddings  $\{\tilde{\mathbf{h}}_u\}$  for all  $u \in \mathcal{U}$ ; obtain item semantic embeddings  $\{\tilde{\mathbf{e}}_v\}$  for all  $v \in \mathcal{V}$ .
  - 2: Freeze the embeddings  $\{\tilde{\mathbf{h}}_u\}$  and  $\{\tilde{\mathbf{e}}_v\}$ , and initialize the model parameters.
  - 3: **for** each iteration **do**
  - 4:   Compute  $\mathbf{h}_u$  using Equation (2).
  - 5:   Calculate  $\hat{\mathbf{y}} = \text{softmax}(\mathbf{h}_u \mathbf{M}^T)$ .
  - 6:   Compute  $\mathcal{L}_{\text{Rec}}$  using Equation (3).
  - 7:   Retrieve  $\mathcal{N}_u$  for each  $u \in \mathcal{U}$  using Equation (5).
  - 8:   Synthesize  $\mathbf{h}_u^+$  for each  $u$  using Equations (6), (7), and (8).
  - 9:   Compute  $\mathcal{L}_{\text{CS}}$  using Equation (9).
  - 10:   Retrieve  $\mathcal{N}_v$  for each  $v \in \mathcal{V}$  using Equation (10).
  - 11:   Generate  $\mathcal{S}'_u$  and  $\mathcal{S}''_u$ , along with the corresponding  $\mathbf{h}'_u$  and  $\mathbf{h}''_u$ .
  - 12:   Compute  $\mathcal{L}_{\text{IS}}$  using Equation (11).
  - 13:   Calculate the total loss  $\mathcal{L} = \mathcal{L}_{\text{Rec}} + \alpha \mathcal{L}_{\text{CS}} + \beta \mathcal{L}_{\text{IS}}$ .
  - 14:   Update the model parameters using the gradient of  $\mathcal{L}$ .
  - 15: **end for**
  - 16: Return the final model parameters  $\theta$ .
- 

**Algorithm 2** Inference for SRA-CL

---

**Require:** Trained model parameters  $\theta$ ; test data  $\{\mathcal{S}_u\}$

- 1: **for** each user sequence in test data **do**
  - 2:   Compute  $\mathbf{h}_u$  using Equation (2).
  - 3:   Calculate the predicted scores  $\hat{\mathbf{y}} = \text{softmax}(\mathbf{h}_u \mathbf{M}^T)$ .
  - 4:   Obtain the top- $k$  items with the highest scores in  $\hat{\mathbf{y}}$ .
  - 5: **end for**
  - 6: Return the recommended items for all users.
-

### 3.3 Training and Inference

During the training phase, all semantic embeddings are fixed. The training objective consists of three components: the loss of the recommendation model  $\mathcal{L}_{\text{Rec}}$ , which serves as the main loss, and the inter-user contrastive loss  $\mathcal{L}_{\text{CS}}$  and intra-user contrastive loss  $\mathcal{L}_{\text{IS}}$ , which act as regularization terms.

$$\mathcal{L} = \mathcal{L}_{\text{Rec}} + \alpha \mathcal{L}_{\text{CS}} + \beta \mathcal{L}_{\text{IS}}, \quad (12)$$

where  $\alpha$  and  $\beta$  are hyperparameters.

During inference, only the recommendation backbone is utilized. The contrastive learning tasks and LLMs' semantic embeddings are not involved in the inference process. This implies that our framework can be deployed in real-world applications without incurring any additional inference latency from incorporating LLMs. The training and inference processes are detailed in Algorithm 1 and Algorithm 2, respectively.

## 4 Experiments

### 4.1 Experimental Settings

**Datasets.** Following previous studies [21, 34, 22], we conducted experiments on four public real-world datasets: Yelp, Amazon Sports, Beauty, and Office. The statistics for these datasets are presented in Table 3. More details about the datasets are shown in Appendix B.1.

**Evaluation Metrics.** To evaluate the performance of the models, we use widely recognized evaluation metrics: Hit Rate (HR) and Normalized Discounted Cumulative Gain (NDCG), follow previous studies [41, 30, 9, 12]. The leave-one-out strategy is employed, where the last interaction is used for testing, the second-to-last interaction for validation, and the remaining interactions for training. To ensure an unbiased evaluation, we rank the prediction on the whole item set without sampling.

**Baseline Methods.** We compare our method with 13 baseline methods, categorized into three groups: 1) *classical methods* (GRU4Rec [10], SASRec [12], BERT4Rec [27]), 2) *contrastive learning-based methods* (S<sup>3</sup>-Rec [46], CL4SRec [34], CoSeRec [21], ICLRec [3], DuoRec [24], MCLRec [22], ICSRec [23]), and 3) *LLM-based methods* (LRD [35], RLMRec [25], LLM-ESR [20]).

**Implementation Details.** All experiments are conducted with a single 32G V100 GPU. The embedding size is set to 64. We adopt the batch size of 256 and employ the Adam optimizer with a learning rate of 0.001. The dropout rate is set to 0.5 across all datasets. Following previous studies [35], we set the maximum sequence length to 20. The early stopping is applied if the metrics on the validation set do not improve over 10 consecutive epochs. For LLM, we use DeepSeek-V3 by invoking its API. We set the LLM's temperature  $\tau$  to 0 and top- $p$  to 0.001. For the text embedding model  $\mathcal{M}$ , we use the pre-trained RoBERTa from Hugging Face. Note that identical settings are adopted for our method and baselines that involve LLMs and text embeddings to ensure fairness. More implementation details can be found in Appendix B.3.

### 4.2 Comparison Results with Baselines

The comparison results are presented in Table 1. Each experiment was conducted five times, and the average results are reported. SRA-CL consistently outperforms all baseline methods across all datasets, achieving performance improvements of up to 11.82%. The improvements are also confirmed by a paired t-test with a significance level of 0.01. Contrastive learning-based methods generally surpass traditional methods (GRU4Rec, SASRec, BERT4Rec). Among the contrastive learning baselines, MCLRec and ICSRec demonstrate superior performance. However, both methods underperform compared to SRA-CL, as they fail to control the quality of contrastive samples. SRA-CL mitigates this issue by introducing semantic-based retrieval augmentation, thereby improving the quality of contrastive samples and enhancing the overall effectiveness of contrastive learning. Regarding LLM-enhanced baselines, they demonstrate superior results compared to classical methods. However, our proposed SRA-CL achieves significant improvements over these LLM approaches. Unlike existing LLM-based methods, SRA-CL is fundamentally different in motivation—it specifically addresses the limitations in contrastive learning through enhanced construction of positive sample pairs using semantic information.

Table 1: Performance comparison of different methods on four datasets. Bold font indicates the best performance, while underlined values represent the second-best. SRA-CL achieves state-of-the-art results among all methods, as confirmed by a paired t-test with a significance level of 0.01. Due to space constraints, additional metrics (HR@10 and NDCG@10) are provided in Appendix C.2.

Model	Yelp		Sports		Beauty		Office	
	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20
GRU4Rec	0.0639	0.0243	0.0325	0.0129	0.0488	0.0189	0.0956	0.0361
SASRec	0.0899	0.0390	0.0498	0.0216	0.0887	0.0382	0.1329	0.0482
BERT4Rec	0.0913	0.0394	0.0578	0.0241	0.0933	0.0399	0.1436	0.0520
S <sup>3</sup> -Rec	0.0964	0.0443	0.0607	0.0262	0.0994	0.0414	0.1568	0.0571
CL4SRec	0.0923	0.0395	0.0562	0.0235	0.0980	0.0416	0.1297	0.0488
CoSeRec	0.0984	0.0404	0.0638	0.0293	0.1034	0.0487	0.1354	0.0516
ICLRec	0.0974	0.0432	0.0636	0.0284	0.1056	0.0482	0.1513	0.0559
DuoRec	<u>0.1173</u>	0.0493	0.0706	0.0302	0.1224	0.0535	0.1549	0.0653
MCLRec	0.1150	0.0486	<u>0.0736</u>	<u>0.0318</u>	<u>0.1239</u>	<u>0.0536</u>	0.1629	0.0684
ICSRec	0.1165	<u>0.0495</u>	0.0728	0.0304	0.1205	0.0528	<u>0.1643</u>	0.0690
LRD	0.1082	0.0455	0.0589	0.0257	0.0931	0.0402	0.1468	0.0577
RLMRec	0.1125	0.0478	0.0664	0.0298	0.1190	0.0521	0.1532	0.0613
LLM-ESR	0.1061	0.0451	0.0638	0.0277	0.1064	0.0515	0.1425	0.0602
<b>SRA-CL</b>	<b>0.1282</b>	<b>0.0533</b>	<b>0.0823</b>	<b>0.0347</b>	<b>0.1314</b>	<b>0.0568</b>	<b>0.1702</b>	<b>0.0725</b>
Improvement	9.29%	7.68%	11.82%	9.12%	6.05%	5.97%	3.59%	5.07%

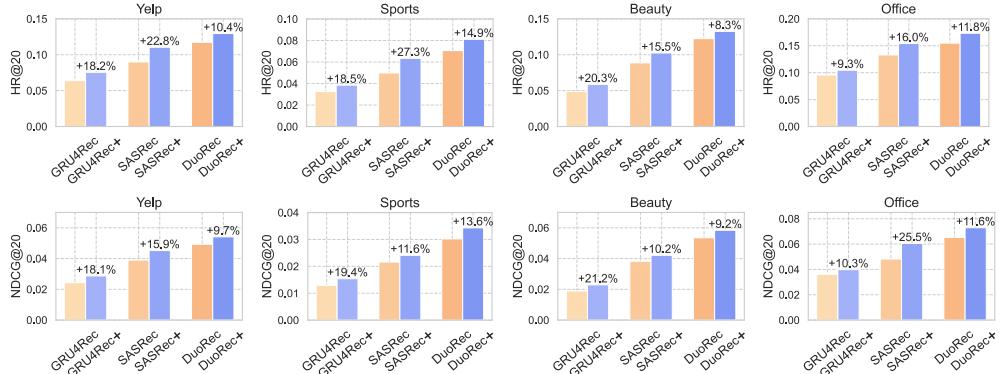


Figure 3: Experimental results demonstrating the model-agnostic nature and strong generalization capability of SRA-CL. “+” indicates the addition of SRA-CL to different recommendation models.

### 4.3 Validation of Model-Agnostic Characteristic

In this section, we validate the model-agnostic nature of our method. We select three classic recommendation models (GRU4Rec, SASRec, DuoRec) as the backbone and integrate SRA-CL to observe performance changes. We retain the original loss functions of the backbones and introduce our contrastive loss  $\mathcal{L}_{CS}$  and  $\mathcal{L}_{IS}$  during training. The results are shown in Figure 3, which indicate that for all three backbone methods, the versions enhanced with SRA-CL (“+”) consistently outperform the original versions. Specifically, HR@20 improves by 8.3% to 27.3%, and NDCG@20 increases by 9.7% to 25.5%. These findings validate that SRA-CL can robustly improve the performance of various recommendation models.

### 4.4 Ablation Study

In this section, we evaluate the effectiveness of each component in SRA-CL. The results, presented in Table 2, demonstrate the impact of removing individual modules. Overall, the results show that removing any component degrades model performance, confirming the necessity of each module. Specifically, the variants “w/o  $\mathcal{L}_{CS}$ ” and “w/o  $\mathcal{L}_{IS}$ ” exhibit significant performance drops, highlighting the importance of both inter-user and intra-user contrastive learning objectives. The “w/o CL” variant suffers a more severe performance decline than those removing only one contrastive objective, suggesting that these two types of objectives complement each other. Additionally, the “w/o learn.” variant also leads to reduced performance, indicating that a learning-based sample synthesizer is more effective than random selection for inter-user contrastive learning. Furthermore, removing

Table 2: Ablation study on all datasets.

	Metric	w/o CL	w/o $\mathcal{L}_{CS}$	w/o $\mathcal{L}_{IS}$	w/o learn.	w/o semantic	w/o LLM	Ours
Yelp	H@20	0.1101	0.1203	0.1228	0.1253	0.1187	0.1190	<b>0.1282</b>
	N@20	0.0473	0.0504	0.0519	0.0520	0.0495	0.0501	<b>0.0533</b>
Sports	H@20	0.0745	0.0780	0.0795	0.0792	0.0772	0.0781	<b>0.0823</b>
	N@20	0.0296	0.0315	0.0332	0.0336	0.0311	0.0314	<b>0.0347</b>
Beauty	H@20	0.1206	0.1273	0.1279	0.1273	0.1265	0.1259	<b>0.1314</b>
	N@20	0.0518	0.0546	0.0545	0.0551	0.0532	0.0537	<b>0.0568</b>
Office	H@20	0.1476	0.1621	0.1619	0.1617	0.1624	0.1643	<b>0.1702</b>
	N@20	0.0599	0.0691	0.0689	0.0681	0.0673	0.0692	<b>0.0725</b>

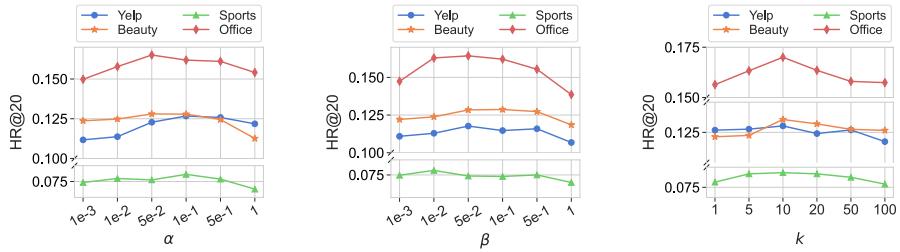


Figure 4: Hyperparameter experiments on the weight of  $\mathcal{L}_{CS}$  ( $\alpha$ ), the weight of  $\mathcal{L}_{IS}$  ( $\beta$ ), and the number of retrieved users/items ( $k$ ).

semantic information and relying solely on collaborative signals for retrieval (“w/o semantic”) results in a notable performance decline, underscoring the importance of semantic information in constructing high-quality contrastive samples. This finding aligns with our initial motivation. Similarly, the absence of LLM-based text processing (“w/o LLM”) also results in performance degradation, demonstrating that utilizing the LLM’s ability to understand and reason about user preferences is crucial.

#### 4.5 Hyperparameter Study

In this section, we investigate the impacts of three key hyperparameters,  $\alpha$ ,  $\beta$ , and  $k$ . Here,  $\alpha$  and  $\beta$  are the weights of  $\mathcal{L}_{CS}$  and  $\mathcal{L}_{IS}$ , respectively, while  $k$  denotes the number of retrieved users/items. From Figure 4, we observe that as both  $\alpha$  and  $\beta$  increase, the model’s performance initially improves slightly and then decreases marginally. Empirically, the optimal range for  $\alpha$  and  $\beta$  is between 0.05 and 0.1. This is reasonable as contrastive learning loss acts as a regularization term. As the value of  $k$  increases, the performance initially improves and then declines, with the optimal value around 10. As  $k$  increases, the semantic relevance of retrieved neighbors decreases and randomness increases. A very small  $k$  results in a candidate set that is too small without diversity. Conversely, a very large  $k$  loses semantic relevance, thereby degrading the effectiveness of contrastive learning. Note that NDCG@20 results are provided in Figure 7 due to space limitation.

#### 4.6 Contrastive Learning in Sparse Data: Analyzing SRA-CL’s Superiority

To further examine SRA-CL’s capability in mitigating the issue of low-quality contrastive samples in data-sparse scenarios, we categorize user sequences into three groups based on their length and compare the evaluation results of different methods. Due to space limitation, we present the experimental results for Beauty and Office, as shown in Figure 5. By comparing SRA-CL with the two strongest contrastive learning baselines (MCLRec and ICSRec), we observe that SRA-CL consistently outperforms them across all user groups. Notably, our method achieves greater improvements in sparser user groups (e.g., those with fewer than 7 or 7-10 interactions). This result further validates our motivation: while MCLRec and ICSRec

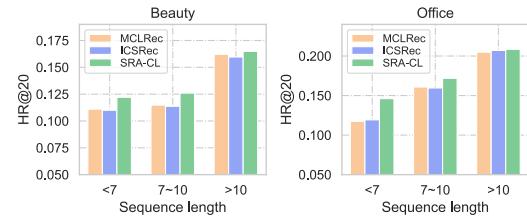


Figure 5: Performance comparison on different user groups among MCLRec, ICSRec and Ours.

construct contrastive sample pairs based on collaborative signals, their performance degrades in data-sparse scenarios due to the diminished quality of contrastive samples. In contrast, our method significantly enhances the quality of contrastive pairs by incorporating semantic information, leading to superior performance under sparse data conditions.

## 5 Related Work

**Contrastive Learning in Sequential Recommendation.** Contrastive learning has been successfully used to enhance sequential recommendation [40, 11, 38, 31, 33, 37, 4, 5, 23]. In terms of the composition of contrastive samples, we categorize existing methods into two types: (1) Inter-user. This involves generating contrastive samples from different user sequences. For example, ICLRec [3] clusters user interests into distinct categories by K-Means and brings the representations of users with similar interests closer. ICSRec [23] further segments a user’s behavior sequence into multiple subsequences to generate finer-grained user intentions for contrastive learning. These methods generate contrastive supervision signals based on collaborative signals. However, the sparsity of the co-occurrence pattern leads to unreliable clustering results, which in turn affects the performance of contrastive learning. (2) Intra-user. This involves applying perturbations to the original sequence to generate augmented views. The two views of the same sequence are treated as a pair of positive samples. For example, CL4SRec [34] employs three data-level augmentation operators: Cropping, Masking, and Reordering, to create contrastive pairs. CoSeRec [21] introduces two additional informative augmentation operators, building upon the foundation of CL4SRec. In addition, some methods generate augmented views from the model’s hidden layers. A notable example is DuoRec [24], which creates positive pairs by forward-passing a sequence representation twice with different dropout masks. MCLRec [22] further combines data-level and model-level augmentation. Despite their effectiveness, they employ random operators, introducing significant uncertainty and potentially generating unreasonable positive samples for contrastive learning.

**Sequential Recommendation with LLMs.** Building upon foundations laid by traditional recommender systems [9, 15, 16], recent studies have successfully integrated LLMs into the recommender paradigm [32, 43, 14, 6]. Overall, LLMs are employed either as direct recommenders or as tools for extracting semantic information [13, 19, 42, 1, 26]. In the former approach, all inputs are converted into textual format, and the LLM generates recommendations based on its pre-trained knowledge or after undergoing supervised fine-tuning. Representative examples include LC-Rec [44], LLM-TRSR [45], and CALRec [18]. However, these methods rely on the inference process of large language models to generate recommendation results, which is computationally expensive and often challenging to deploy in practical scenarios. Another line of research [26, 30, 20, 35, 42, 1] leverages LLMs to process semantic information and incorporates it into traditional ID-based models. For example, SLIM [30] distills knowledge from large-scale LLMs into a smaller student LLM to improve the recommendation model. LLM-ESR [20] addresses the long-tail problem by leveraging collaborative signals and semantic information through dual-view modeling and self-distillation. LRD [35] utilizes the LLM to explore potential relations between items and reconstructs one item based on its relation to another. Unlike the aforementioned methods, our approach, grounded in the essence of contrastive learning, aims to construct more effective contrastive pairs with LLMs.

## 6 Conclusion

In this paper, we analyze the limitations of contrastive learning in sequential recommendation, namely Semantic Divergence and Unlearnability. To address these issues, we propose SRA-CL, a novel framework that enhances contrastive sample construction by integrating LLM-based semantic retrieval with a learnable sample synthesizer. SRA-CL leverages the capabilities of LLMs without increasing the inference time of the recommendation model, making it practical for large-scale real-world applications. Through comprehensive experiments, we demonstrate that LLM-based semantic-guided contrastive sample construction improves the contrastive learning, and we validate the effectiveness of the learnable sample synthesis mechanism. Furthermore, experiments with different recommendation model backbones confirm the model-agnostic nature of our approach.

## 7 Acknowledgements

This work was supported by the Early Career Scheme (No. CityU 21219323) and the General Research Fund (No. CityU 11220324) of the University Grants Committee (UGC), the NSFC Young Scientists Fund (No. 9240127), and the Donation for Research Projects (No. 9220187 and No. 9229164).

## References

- [1] Artun Boz, Wouter Zorgdrager, Zoe Kotti, Jesse Harte, Panos Louridas, Dietmar Jannach, and Marios Fragnoulis. Improving sequential recommendations with llms. *arXiv preprint arXiv:2402.01339*, 2024.
- [2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [3] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 2172–2182, 2022.
- [4] Ziqiang Cui, Yixin Su, Fangquan Lin, Cheng Yang, Hanwei Zhang, and Jihai Zhang. Dual disentangled attention for multi-information utilization in sequential recommendation. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.
- [5] Ziqiang Cui, Haolun Wu, Bowei He, Ji Cheng, and Chen Ma. Context matters: Enhancing sequential recommendation with context-aware diffusion-based contrastive learning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 404–414, 2024.
- [6] Ziqiang Cui, Yunpeng Weng, Xing Tang, Fuyuan Lyu, Dugang Liu, Xiuqiang He, and Chen Ma. Comprehending knowledge graphs with large language models for recommender systems. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1229–1239, 2025.
- [7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [8] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.
- [9] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgc: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.
- [10] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.
- [11] Yangqin Jiang, Yuhao Yang, Lianghao Xia, and Chao Huang. Diffkg: Knowledge graph diffusion model for recommendation. *arXiv preprint arXiv:2312.16890*, 2023.
- [12] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.
- [13] Lei Li, Yongfeng Zhang, and Li Chen. Prompt distillation for efficient llm-based recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1348–1357, 2023.
- [14] Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. Large language models for generative recommendation: A survey and visionary discussions. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10146–10159, 2024.
- [15] Shiwei Li, Huifeng Guo, Lu Hou, Wei Zhang, Xing Tang, Ruiming Tang, Rui Zhang, and Ruixuan Li. Adaptive low-precision training for embeddings in click-through rate prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4435–4443, 2023.

- [16] Shiwei Li, Zhuoqi Hu, Xing Tang, Haozhao Wang, Shijie Xu, Weihong Luo, Yuhua Li, Xiuqiang He, and Ruixuan Li. Mixed-precision embeddings for large-scale recommendation models. *arXiv preprint arXiv:2409.20305*, 2024.
- [17] Xuewei Li, Aitong Sun, Mankun Zhao, Jian Yu, Kun Zhu, Di Jin, Mei Yu, and Ruigu Yu. Multi-intention oriented contrastive learning for sequential recommendation. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, pages 411–419, 2023.
- [18] Yaoyiran Li, Xiang Zhai, Moustafa Alzantot, Keyi Yu, Ivan Vulić, Anna Korhonen, and Mohamed Hammad. Calrec: Contrastive alignment of generative llms for sequential recommendation. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 422–432, 2024.
- [19] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. Llara: Large language-recommendation assistant. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1785–1795, 2024.
- [20] Qidong Liu, Xian Wu, Yeqing Wang, Zijian Zhang, Feng Tian, Yefeng Zheng, and Xiangyu Zhao. Llm-esr: Large language models enhancement for long-tailed sequential recommendation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [21] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S Yu, Julian McAuley, and Caiming Xiong. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479*, 2021.
- [22] Xiuyuan Qin, Huanhuan Yuan, Pengpeng Zhao, Junhua Fang, Fuzhen Zhuang, Guanfeng Liu, Yanchi Liu, and Victor Sheng. Meta-optimized contrastive learning for sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 89–98, 2023.
- [23] Xiuyuan Qin, Huanhuan Yuan, Pengpeng Zhao, Guanfeng Liu, Fuzhen Zhuang, and Victor S Sheng. Intent contrastive learning with cross subsequences for sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 548–556, 2024.
- [24] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 813–823, 2022.
- [25] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. Representation learning with large language models for recommendation. In *Proceedings of the ACM Web Conference 2024*, pages 3464–3475, 2024.
- [26] Yankun Ren, Zhongde Chen, Xinxing Yang, Longfei Li, Cong Jiang, Lei Cheng, Bo Zhang, Linjian Mo, and Jun Zhou. Enhancing sequential recommenders with augmented knowledge from aligned large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 345–354, 2024.
- [27] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019.
- [28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [29] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

- [30] Yuling Wang, Changxin Tian, Binbin Hu, Yanhua Yu, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, Liang Pang, and Xiao Wang. Can small language models be good reasoners for sequential recommendation? In *Proceedings of the ACM on Web Conference 2024*, pages 3876–3887, 2024.
- [31] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 726–735, 2021.
- [32] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. A survey on large language models for recommendation. *World Wide Web*, 27(5):60, 2024.
- [33] Lianghao Xia, Chao Huang, Chunzhen Huang, Kangyi Lin, Tao Yu, and Ben Kao. Automated self-supervised learning for recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 992–1002, 2023.
- [34] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 1259–1273. IEEE, 2022.
- [35] Shenghao Yang, Weizhi Ma, Peijie Sun, Qingyao Ai, Yiqun Liu, Mingchen Cai, and Min Zhang. Sequential recommendation with latent relations based on large language model. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–344, 2024.
- [36] Yuhao Yang, Chao Huang, Lianghao Xia, Chunzhen Huang, Da Luo, and Kangyi Lin. Debiased contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 1063–1073, 2023.
- [37] Yihang Yin, Qingzhong Wang, Siyu Huang, Haoyi Xiong, and Xiang Zhang. Autogcl: Automated graph contrastive learning via learnable view generators. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 8892–8900, 2022.
- [38] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 1294–1303, 2022.
- [39] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. Self-supervised learning for recommender systems: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(1):335–355, 2023.
- [40] Dan Zhang, Yangliao Geng, Wenwen Gong, Zhongang Qi, Zhiyu Chen, Xing Tang, Ying Shan, Yuxiao Dong, and Jie Tang. Recdcl: Dual contrastive learning for recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3655–3666, 2024.
- [41] Xiaokun Zhang, Bo Xu, Youlin Wu, Yuan Zhong, Hongfei Lin, and Fenglong Ma. Finerec: Exploring fine-grained sequential recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1599–1608, 2024.
- [42] Yuyue Zhao, Jiancan Wu, Xiang Wang, Wei Tang, Dingxian Wang, and Maarten de Rijke. Let me do it for you: Towards llm empowered recommendation via tool learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1796–1806, 2024.
- [43] Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, et al. Recommender systems in the era of large language models (llms). *arXiv preprint arXiv:2307.02046*, 2023.

- [44] Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 1435–1448. IEEE, 2024.
- [45] Zhi Zheng, Wenshuo Chao, Zhaopeng Qiu, Hengshu Zhu, and Hui Xiong. Harnessing large language models for text-rich sequential recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3207–3216, 2024.
- [46] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1893–1902, 2020.
- [47] Peilin Zhou, Jingqi Gao, Yueqi Xie, Qichen Ye, Yining Hua, Jaeboum Kim, Shoujin Wang, and Sunghun Kim. Equivariant contrastive learning for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 129–140, 2023.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction provide a concise yet comprehensive overview of our core innovation, key methodological contributions, and the specific research problem addressed in this work.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We discuss the limitations of the work in the appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: This paper provides full disclosure of all information necessary to reproduce our key experimental results, and provides source code on an anonymized GitHub repository.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

#### 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: The four datasets used in this paper are all public, and we offer their links. We also provide our source code on the anonymized GitHub repository.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: We specify all training and test details, including data splits, hyperparameters, how they were chosen, type of optimizer, etc. necessary to understand the results in the experiment section and the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[Yes\]](#)

Justification: The experimental results in this paper are confirmed by a paired t-test with a significance level of 0.01.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer “Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: **[Yes]**

Justification: The paper provides sufficient information on the computer resources needed to reproduce the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: **[Yes]**

Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

## 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: **[Yes]**

Justification: The paper discussed both potential positive societal impacts and negative societal impacts in the appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: There is no risk of misuse of the proposed method and the datasets used in the paper are open-sourced.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All third-party assets are properly credited through citations and in-text acknowledgments.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](http://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: We communicate the details of the dataset/code/model as part of our submission with an anonymized github URL.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

#### 16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

## A Technical Supplement to SRA-CL

### A.1 Prompt Template

In this section, we provide a detailed description of the LLM prompt templates employed in our study. Specifically, to enhance the model’s ability to comprehend user preferences and items, we have meticulously designed specialized prompts, as illustrated in Figure 6.

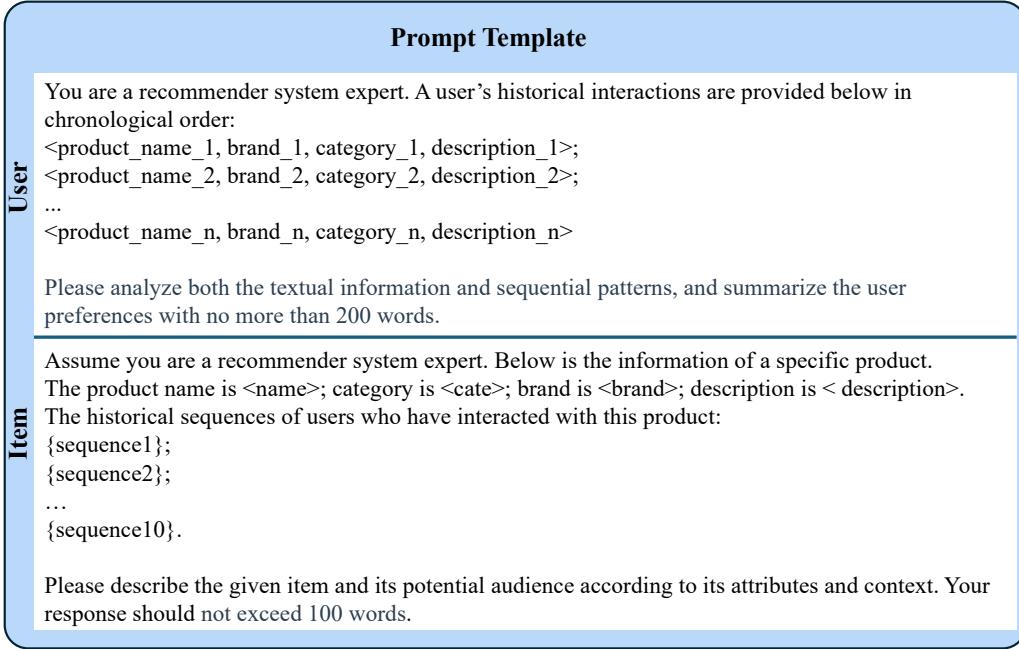


Figure 6: Prompt Template.

### A.2 Efficiency Analysis

**Inference Efficiency.** During inference, only the recommendation backbone is utilized. The contrastive learning tasks and the LLMs’ semantic embeddings are not involved in the inference process. This ensures that our framework can be deployed in real-world applications without incurring any additional inference latency from incorporating LLMs.

**Training Efficiency.** The training process of our method consists of two stages: In the first stage, we use an LLM API to obtain semantic information and convert it into embeddings, which are then cached to construct contrastive sample indices. The primary time cost in this stage comes from the API calls. However, by employing asynchronous concurrency, this step can be completed within a few hours. Crucially, this stage is performed once and requires no repetition during model training. In the second stage, we use the pre-constructed contrastive sample index to train the recommendation model. Regarding the computational complexity of this stage, our method maintains comparable time complexity to general ID-based contrastive recommendation approaches. The only additional overhead during training compared to conventional contrastive recommendation models comes from the lightweight learnable sample synthesis module whose parameter size is negligible compared to that of the main recommendation model.

## B Experimental Setting Details

### B.1 Datasets

We conducted experiments on four public real-world datasets: Yelp, Sports, Beauty, and Office. The statistics for these datasets are presented in Table 3. These datasets cover a diverse range of

Table 3: Dataset statistics.

Datasets	#Users	#Items	#Actions	Avg. Length	Density
Yelp	19,936	14,587	207,952	10.4	0.07%
Sports	35,598	18,357	296,337	8.3	0.05%
Beauty	22,363	12,101	198,502	8.8	0.07%
Office	4,905	2,420	53,258	10.9	0.45%

application scenarios. The Yelp dataset<sup>3</sup> is widely used for business recommendations. The Sports, Beauty, and Office datasets are sourced from Amazon<sup>4</sup>, one of the largest e-commerce platforms. Following previous studies [21, 34, 22], the users and items that have fewer than five interactions are removed.

## B.2 Baseline Methods

To ensure a comprehensive assessment, we compare our method with 12 baseline methods, categorized into three groups: classical methods (GRU4Rec, SASRec, BERT4Rec), contrastive learning-based methods (S<sup>3</sup>-Rec, CL4SRec, CoSeRec, ICLRec, DuoRec, MCLRec, ICSRec), and LLM-enhanced methods (LRD, LLM-ESR).

- **GRU4Rec** [10] applies recurrent neural networks (RNN) to sequential recommendation.
- **SASRec** [12] is the first work to utilize the self-attention mechanism for sequential recommendation.
- **BERT4Rec** [27] employs the BERT [7] framework to capture the context information of user behaviors.
- **S<sup>3</sup>-Rec** [46] leverages four self-supervised objectives to uncover the inherent correlations within the data.
- **CL4SRec** [34] proposes three random augmentation operators to generate positive samples for contrastive learning.
- **CoSeRec** [21] introduces two additional informative augmentation operators, building upon the foundation of CL4SRec.
- **ICLRec** [3] clusters user interests into distinct categories and brings the representations of users with similar interests closer together.
- **DuoRec** [24] combines a model-level dropout augmentation and a sampling strategy for choosing hard positive samples.
- **MCLRec** [22] integrates CL4SRec’s random data augmentation for the input sequence and employs MLP layers for model-level augmentation.
- **ICSRec** [23] is an improvement on ICLRec, further segmenting a user’s sequential behaviors into multiple subsequences to generate finer-grained user intentions for contrastive learning.
- **LRD** [35] is an LLM-based method. It leverages LLMs to discover new relations between items and reconstructs one item based on the relation and another item.
- **RLMRec** [25] utilizes LLMs to generate text profiles and combine their semantic embeddings with recommendation models.
- **LLM-ESR** [20] is also an LLM-based method. It addresses the long-tail problem by simultaneously leveraging collaborative signals and semantic information through the dual-view modeling and self-distillation.

## B.3 Implementation Details

All experiments are conducted with a single V100 GPU. The embedding size for all methods is set to 64 for a fair comparison. We use a training batch size of 256 and employ the Adam optimizer

<sup>3</sup><https://www.yelp.com/dataset>

<sup>4</sup><http://jmcauley.ucsd.edu/data/amazon/>

with a learning rate of 0.001. The dropout rate is set to 0.5 for both the embedding layer and the hidden layers across all datasets. Following previous studies [35], we set the maximum sequence length to 20 for all datasets. The early stopping is applied if the metrics on the validation set do not improve over 10 consecutive epochs. Our method is model-agnostic and can be applied to any sequential recommendation model. The transformer backbone mentioned in Sec. 2.2 comprises two layers, each with two attention heads. For the LLM, we select DeepSeek-V3, a robust large language model that demonstrates exceptional performance on both standard benchmarks and open-ended generation evaluations. For detailed information about DeepSeek, please refer to their official website<sup>5</sup>. Specifically, we utilize DeepSeek-V3 by invoking its API<sup>6</sup>. To reduce text randomness of the LLM, we set the temperature  $\tau$  to 0 and the top- $p$  to 0.001. For the text embedding model  $\mathcal{M}$ , we use the pre-trained SimCSE-RoBERTa<sup>7</sup> from Hugging Face. Identical settings are used for baselines that involve LLMs and text embeddings to ensure fairness.

## C Additional Results & Analysis

### C.1 Discussion on Learnable Sample Synthesis

**Inter-User Contrastive Learning.** User preferences exhibit significant heterogeneity across individuals. Sole reliance on hard rules, such as selecting a user from the current user’s dedicated candidate pool as the positive sample, may yield suboptimal solutions. Our experiments (as shown in Table 2 “w/o learn.”) validated this. To enhance contrastive sample construction, we introduce a learnable sample synthesizer that optimizes the contrastive sample generation process during model training for inter-user contrastive learning.

**Intra-User Contrastive Learning.** Our preliminary experiments also explored the use of learnable synthesizers (analogous to inter-contrastive learning approaches) for generating substitute items, yet yielded no measurable performance improvements (shown in Table 4). Our analysis suggests this results from the inherent nature of item semantics being more readily interpretable and quantifiable than user preferences. Therefore, directly identifying appropriate substitutes from semantically similar candidate pools is simpler and more reliable compared to matching users with analogous preference patterns.

Table 4: Performance impact of learnable versus non-learnable sample synthesis strategies in **intra-user** contrastive learning.

	Yelp		Sports		Beauty		Office	
	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20
Learnable	0.1276	0.0531	0.0825	0.0344	0.1309	0.0561	0.1706	0.0722
Unlearnable	0.1282	0.0533	0.0823	0.0347	0.1314	0.0568	0.1702	0.0725

### C.2 Additional Comparison Results

We provide additional comparison results (HR@10 and NDCG@10) of different methods in Table 5. The experimental results demonstrate that our method outperforms all baselines across all datasets, further validating its superiority.

### C.3 Additional Results for Hyperparameter Experiments

Due to space constraints, we only present HR@20 in Figure 4 of the main text for hyperparameter study. Here, we additionally report the NDCG@20 evaluation results in Figure 7, providing complementary performance metrics for comprehensive analysis. As shown, the trend in NDCG@20 closely aligns with that of HR@20.

<sup>5</sup><https://github.com/deepseek-ai/DeepSeek-V3>

<sup>6</sup><https://api-docs.deepseek.com/>

<sup>7</sup><https://huggingface.co/princeton-nlp/sup-simcse-roberta-large>

Table 5: Additional comparison results for HR@10 and NDCG@10. Bold font indicates the best performance, while underlined values represent the second-best. “ND” represents for “NDCG”. Our method SRA-CL achieves state-of-the-art results among all methods, as confirmed by a paired t-test with a significance level of 0.01.

Model	Yelp		Sports		Beauty		Office	
	HR@10	ND@10	HR@10	ND@10	HR@10	ND@10	HR@10	ND@10
GRU4Rec	0.0362	0.0173	0.0193	0.0096	0.0279	0.0137	0.0540	0.0260
SASRec	0.0572	0.0308	0.0304	0.0157	0.0612	0.0336	0.0791	0.0348
BERT4Rec	0.0582	0.0311	0.0349	0.0189	0.0628	0.0352	0.0821	0.0376
S <sup>3</sup> -Rec	0.0612	0.0339	0.0385	0.0204	0.0647	0.0327	0.0931	0.0426
CL4SRec	0.0583	0.0315	0.0358	0.0189	0.0649	0.0329	0.0695	0.0322
CoSeRec	0.0607	0.0309	0.0439	0.0244	0.0725	0.0410	0.0782	0.0412
ICLRec	0.0598	0.0328	0.0428	0.0235	0.0713	0.0396	0.0922	0.0411
DuoRec	0.0747	0.0380	0.0474	0.0242	0.0841	0.0443	0.1015	0.0519
MCLRec	0.0721	0.0378	0.0498	0.0257	0.0870	0.0442	0.1036	0.0538
ICSRec	0.0738	0.0380	0.0487	0.0243	0.0844	0.0437	0.1034	0.0540
LRD	0.0693	0.0357	0.0376	0.0191	0.0620	0.0294	0.0887	0.0431
RLMRec	0.0709	0.0371	0.0426	0.0238	0.0764	0.0439	0.0927	0.0496
LLM-ESR	0.0669	0.0353	0.0415	0.0221	0.0750	0.0435	0.0889	0.0468
<b>SRA-CL</b>	<b>0.0817</b>	<b>0.0419</b>	<b>0.0539</b>	<b>0.0274</b>	<b>0.0924</b>	<b>0.0469</b>	<b>0.1111</b>	<b>0.0575</b>
Improvement	9.37%	10.26%	8.23%	6.61%	6.21%	6.11%	7.24%	6.48%

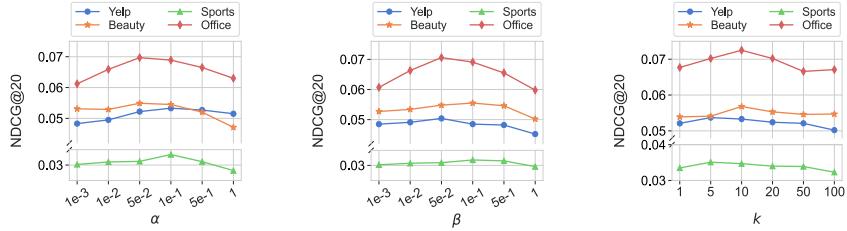


Figure 7: Hyperparameter experiments on the weight of  $\mathcal{L}_{CS}$  ( $\alpha$ ), the weight of  $\mathcal{L}_{IS}$  ( $\beta$ ), and the number of retrieved users/items ( $k$ ) (NDCG results).

## D Other Discussions

### D.1 Limitation

Considering computational budgets and resource limitations, we specifically analyzed how two selected LLMs (DeepSeek and Qwen) affect our framework’s effectiveness. While more LLMs might yield different results, our study focused on these representative models.

### D.2 Broader Impacts

SRA-CL demonstrates significant improvements in sequential recommendation accuracy (positive impact), with potential applicability to real-world platforms. Like all recommendation systems, its personalized nature may occasionally limit content diversity, though this effect is inherent to the recommendation paradigm rather than unique to our method.