CLLMRec: Contrastive Learning with LLMs-based View Augmentation for Sequential Recommendation

Fan Lu[1], **Xiaolong Xu**[1*], **Haolong Xiang**[1], **Lianyong Qi**[2], **Xiaokang Zhou**[3], **Fei Dai**[4] and **Wanchun Dou**[5]
[1] Nanjing University of Information Science and Technology
[2] China University of Petroleum (East China)
[3] Kansai University
[4] Southwest Forestry University
[5] Nanjing University
{fanl, xlxu, hlxiang}@nuist.edu.cn, {lianyongqi, flydai.cn}@gmail.com, zhou@kansai-u.ac.jp, douwc@nju.edu.cn

Abstract

Sequential recommendation generates embedding representations from historical user-item interactions to recommend the next potential interaction item. Due to the complexity and variability of historical user-item interactions, extracting effective user features is quite challenging. Recent studies have employed sequential networks such as time series networks and Transformers to capture the intricate dependencies and temporal patterns in historical user-item interactions, extracting more effective user features. However, limited by the scarcity and suboptimal quality of data, these methods struggle to capture subtle differences in user sequences, which results in diminished recommendation accuracy. To address the above issue, we propose a contrastive learning framework with LLMs-based view augmentation (CLLMRec), which effectively mines differences in behavioral sequences through sample generation. Specifically, CLLMRec utilizes LLMs (Large Language Models) to augment views and expand user behavior sequence representations, providing high-quality positive and negative samples. Subsequently, CLLMRec employs the augmented views for effective contrastive learning, capturing subtle differences in behavioral sequences to suppress interference from irrelevant noise. Experimental results on three public datasets demonstrate that the proposed method outperforms state-of-the-art baseline models, and significantly enhances recommendation performance.

1 Introduction

Sequential recommendation (SR) plays a crucial role in various practical application scenarios, such as purchase prediction [Liang *et al.*, 2024], web page recommendation [Zhang *et al.*, 2024a], and next point-of-interest recommendation [Chen *et al.*, 2024]. SR necessitates a thorough analysis of historical user behavior data to operate effectively, consequently relying heavily on user preference data [Yang *et al.*, 2024b]. Due to the rapid changes in user preferences and the continual increase in interaction data volume, SR faces significant challenges in handling complex and dynamic user interaction data [Qin *et al.*, 2024].

Current research typically utilizes neural network models to identify feature representations of highly dynamic data [Zhao *et al.*, 2024]. Early approaches utilized Convolutional Neural Networks and Recurrent Neural Networks to process user behavior data, capturing temporal features and local visual characteristics of user actions [Zhang *et al.*, 2024b]. With technological advancements, models based on attention mechanisms, such as the Transformer, have become a popular choice for handling dynamic data [Xu *et al.*, 2025]. These models enhance the capability for feature identification by concentrating on pivotal information within sequences and effectively handle long sequences of dynamic data.

However, due to the scarcity and suboptimal quality of interaction data, these methods struggle to accurately capture the subtle differences in user preferences. This primarily stems from insufficient user interaction data to construct reliable user behavior models, which may result in inaccuracies in the learned features [Li *et al.*, 2024]. Additionally, these methods overlook the interference from irrelevant noise, limiting the overall performance of the recommendation systems [Zhang *et al.*, 2024c]. Therefore, effectively enhancing the quantity and quality of data, while suppressing the interference of irrelevant noise, is crucial for improving the performance of SR.

To address the above challenges, we propose a **C**ontrastive **L**earning Framework with L**LM**s-based View Augmentation for sequential **Rec**ommendation (CLLMRec). The framework utilizes pre-trained LLMs to optimize view enhancement strategies, producing high-quality contrastive views. Subsequently, it employs contrastive learning to extract distinctions within behavior sequences and suppress the interference of irrelevant noise. Additionally, CLLMRec exhibits

*Corresponding Author

strong modularity, making it easy to integrate with existing recommender system architectures, serving as a plug-and-play module for various SR tasks. The contributions of this paper are as follows:

- We proposed CLLMRec, a contrastive learning framework with LLMs-based view augmentation. By leveraging pre-trained LLMs to automatically evaluate data importance and adaptively generate high-value contrastive views based on recommendation scenarios, CLLMRec enhances recommendation performance and is theoretically compatible with all SR models.
- We use self-supervised contrastive learning in CLLMRec to capture differences between samples, effectively reducing the impact of irrelevant noise. This approach enhances the recommendation model's ability to recognize complex user interactions, significantly improving the performance and generalization capability of the recommendation.
- The framework was evaluated on three publicly available datasets, and the results demonstrate that it outperforms state-of-the-art models in all scenarios. Additionally, further ablation experiments validate the effectiveness of the LLMs-based view augmentation method and the contrastive learning module.

2 Related Work

2.1 LLMs-Based Recommendation Algorithms

Recommender systems based on LLMs have become a research focus, with various approaches integrating linguistic and collaborative semantics to boost performance. Zheng et al. [Zheng *et al.*, 2024] proposed LC-Rec, which combines language and collaborative semantics using vector quantization and fine-tuning tasks, improving recommendation accuracy. Lin et al. [Lin *et al.*, 2024] introduced TransRec, enhancing product representation by balancing semantics and distinctiveness through product IDs, titles, and attributes, and improving identifier generation with an aggregation module.

Addressing fairness, Jiang et al. [Jiang *et al.*, 2024] developed the IFairLRS framework, which minimizes historical biases in recommendations, ensuring fairness across user groups. Liao et al. [Liao *et al.*, 2024] proposed a hybrid strategy combining traditional recommendations with LLM knowledge, improving SR performance and partially addressing the cold-start problem. Zhang et al. [Zhang *et al.*, 2023] approached the task as instruction execution, optimizing personalized recommendations through instruction fine-tuning. In the realm of representation learning, Ren et al. [Ren *et al.*, 2024] introduced RLMRec, which improves user preference modeling with cross-view alignment, expanding recommendation design possibilities. Yang et al. [Yang *et al.*, 2024a] leveraged LLMs to uncover latent product relationships, optimizing SR via self-supervised tasks. These studies highlight the potential of integrating linguistic and collaborative semantics, offering insights for future innovations.

2.2 Contrastive Learning

Contrastive learning-based recommendation methods have significantly improved data sparsity and recommendation quality. Jiang et al. [Jiang *et al.*, 2023] proposed the Adaptive Graph Contrastive Learning (AdaGCL) method, which addresses sparsity and noise in collaborative filtering through an adaptive contrastive view generator. He et al. [He *et al.*, 2023] introduced the Candidate-aware Graph Contrastive Learning (CGCL) method, which improves recommendation accuracy by constructing multiple contrastive learning objectives to enhance node embeddings. To tackle the cold-start problem, Xu et al. [Xu *et al.*, 2024] proposed the Cross-Modal Contrastive Learning-based Cold-Start Recommendation framework (CMCLRec), which generates simulated behavior sequences via cross-modal mapping, thereby enhancing SR models in cold-start scenarios. Yang et al. [Yang *et al.*, 2023] combined generative models with contrastive learning to introduce the Variational Graph Contrastive Learning (VGCL) framework, overcoming the limitations of data augmentation. Hao et al. [Hao *et al.*, 2023] proposed a learnable contrastive learning method that generates contrastive views through model-based augmentation and alleviates supervision sparsity with multi-positive sample contrastive loss.

Despite significant advances in contrastive learning and LLMs-based recommendation methods in handling user features, these approaches still rely on manually designed views and substantial training data to achieve the desired accuracy. We propose the CLLMRec framework, which automates the generation of high-quality contrastive views and utilizes pre-trained LLMs to reduce dependence on large datasets.

3 Preliminary

The SR task focuses on forecasting the items a user might be interested in future moments, using their past behavior sequence as a basis. This problem typically involves time dependencies, dynamic changes in user interests, and the learning of behavior patterns. Therefore, modeling for SR must account for user behavior data across multiple time steps to accurately predict their next choice.

Let the input be the user's behavior sequence across different time steps. Assume that a user u has a series of historical interactions over T time steps, denoted as:

$$S_u = \{(i_t, t) | i_t \in \mathcal{I}, t = 1, 2, ..., T\}, \tag{1}$$

where i_t represents the item (or entity) interacted with by user u at time step t, $\mathcal{I}$ is the set of all items, and t is the time stamp. For each user, $\mathbf{S}_u$ represents the sequence of their interacted items.

The objective is to forecast the item with which the user is most likely to engage at the next time step $T+1$, using their previous behavior sequence as a reference. The output is a recommendation list, represented as a probability distribution over items $\hat{P}(i_{T+1}|\mathbf{S}_u)$, where i_{T+1} is the item the user is most likely to interact with at time step $T+1$.

The model employs gradient descent to adjust the network parameters, aiming to reduce the loss function. The objective function is usually expressed as the negative log-likelihood of the actual next item, conditioned on the historical sequence:

$$\mathcal{L} = -\log P(i_{T+1}|\mathbf{S}_u), \tag{2}$$

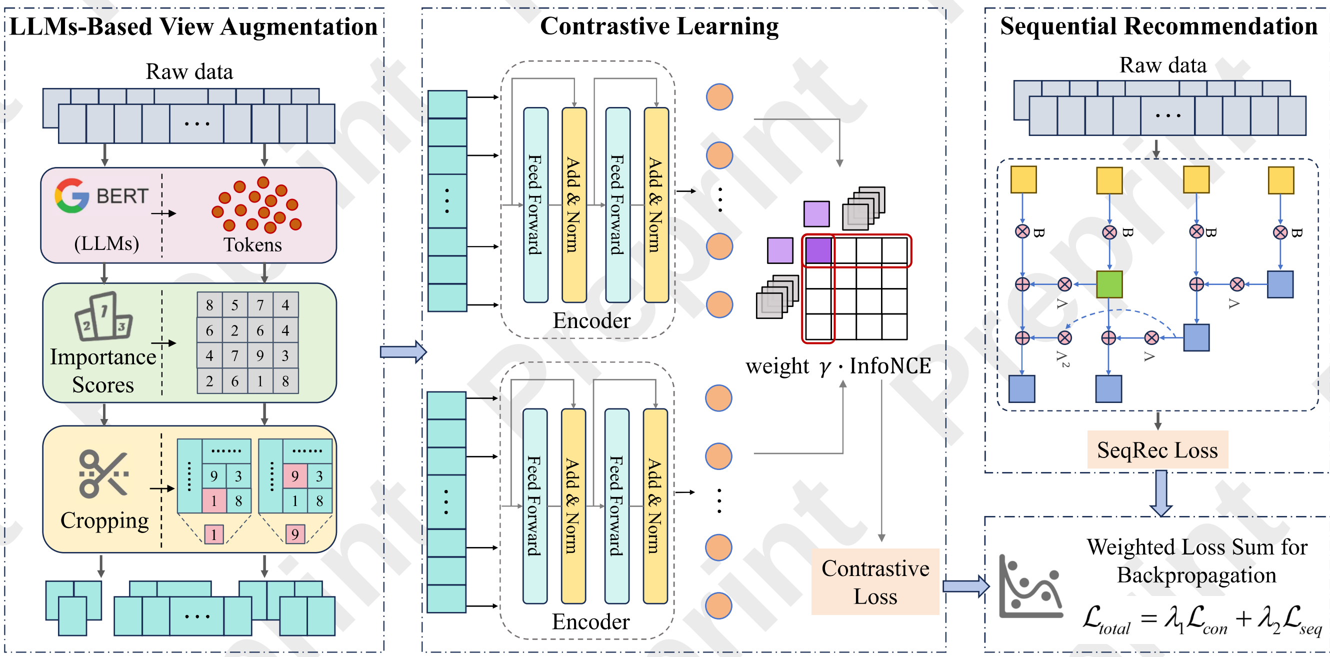

Figure 1: The overall architecture of CLLMRec.

where i_{T+1} represents the actual item the user interacts with at time step $T+1$, and $\hat{P}(i_{T+1}|\mathbf{S}_u)$ denotes the forecasted probability distribution across all potential items. By minimizing this loss, the neural network learns to capture temporal dependencies in the user's behavior sequence and generate personalized recommendations.

4 Methodology

To address the issues of data scarcity and extraneous noise, we introduce the CLLMRec framework, which utilizes LLMs for view data augmentation and employs contrastive learning methods to capture subtle differences in user representations. The CLLMRec framework consists of three modules: the first is the LLMs-based view augmentation module, responsible for generating high-quality contrastive views; the second is the contrastive learning module, which uses these views to further optimize the representations of users, enhancing the effectiveness of recommendations. The third is the sequence recommendation module, which is used for joint training with the contrastive learning module. The complete structure of the framework is illustrated in Fig. 1.

4.1 LLMs-Based View Augmentation Module

In this module, we propose a view enhancement method based on LLMs. The core idea of this method is to calculate an importance score based on the hidden states and self-attention weights at each position. Then, a bidirectional threshold pruning strategy is implemented to prune the parts with lower scores to obtain positive sample views and prune the parts with higher scores to obtain negative sample views, enabling the model to more accurately capture key features while ignoring redundant information. This method relies on the pre-trained LLM BERT, which evaluates the relative importance of each position by analyzing the hidden representations of the input sequence.

Input Sequence and Hidden State. Let the input sequence be represented as $S = \{s_1, s_2, \ldots, s_t\}$, where s_i denotes the i-th element of the sequence, and t is the time step length of the sequence. The sequence S is processed by the pre-trained LLM BERT, and the model outputs a hidden representation for each position. These hidden representations are used for subsequent pruning decisions. After processing by the model, the hidden state is represented as $H = \{h_1, h_2, \ldots, h_t\}$, where $h_i \in \mathbb{R}^d$ is the hidden vector for the i-th position, and d is the dimension of the hidden vector. The hidden state encapsulates the contextual information of each position in the sequence, reflecting the semantic contribution of each position to the entire sequence.

Calculation of Importance Scores. To evaluate the importance of each position, we calculate it by combining the mean of the positional hidden states and their self-attention weights. Specifically, for each position i, its importance score γ_i is computed as follows: First, calculate the absolute mean value of the hidden state h_i at that position, and then integrate the attention weight of position i on itself in the self-attention mechanism. Let $h_i \in \mathbb{R}^d$ represent the hidden state, and $A_{i,i}(h)$ denote the attention weight of position i to itself in the h-th attention head. Then the score γ_i of position i is defined as:

$$\gamma_i = \frac{1}{d}\sum_{j=1}^{d}|h_{ij}| + \lambda_a \cdot \frac{1}{H}\sum_{h=1}^{H} A_{i,i}^{(h)}, \tag{3}$$

where h_{ij} represents the value of the j-th element in the hidden vector h_i, H is the total number of attention heads, and

λ_a is a hyperparameter for balancing the weights of the two terms. The higher the score, the more semantically important the position is and the more focused it is by the model's internal attention, such as sudden changes in user behavior or key points of interest; the lower the score, the smaller the contribution of that position to the model's understanding. By integrating the mean and attention signals, this method can more accurately identify the key positions in the sequence.

Sequence Cropping Strategy. After calculating the importance score γ_i of each position, a clipping strategy based on score ranking is applied. Specifically, the parts with lower scores are clipped to obtain positive samples, and the parts with higher scores are clipped to obtain negative samples. Let the clipping ratio be α; then elements with scores lower than or higher than a certain threshold are selected for clipping. The clipping process can be carried out in the following steps:

- Compute the scores γ_i for all positions.
- Sort the positions by their scores γ_i.
- Select the position according to the score and crop them.

The new contrastive learning views are generated through sequence pruning. By using the pruned data as the new input for contrastive learning, the problem of insufficient data is solved, enabling the model to utilize richer contextual information for training and thus improving the training quality. The pruned positive and negative samples ensure the effectiveness of contrastive learning, avoid interference from irrelevant data, and help the model focus on potentially important features.

4.2 Contrastive Learning Module

In this module, a plug-in contrastive learning method is proposed. This module generates two versions of a sequence by applying two augmentations to the input data and utilizes these sequences for contrastive learning. For each pair of augmented samples, their similarity is calculated based on their representations, and a loss function is constructed according to similarity. The contrastive learning module adopts the Self-Attention-based Sequential Recommendation model as the base model, leveraging the self-attention mechanism to model the user's behavior sequence.

Given a user's behavior sequence $S = \{s_1, s_2, \ldots, s_t\}$, temporal features are extracted from the sequence, and latent space representations are generated using the multi-layer self-attention structure $f(\cdot)$. The input to the model is the user's historical behavior sequence S. After applying the self-attention mechanism, the embedding representation $\mathbf{e}_t$ for each time step is obtained:

$$\mathbf{e}_t = f(s_t), \quad t = 1, 2, \ldots, T, \tag{4}$$

where $\mathbf{e}_t \in \mathbb{R}^d$ is the embedding vector for the t-th time step, d refers to the dimension of the embedding, and T signifies the length of the sequence. For a specified input sequence S, the sequence's final representation is derived from the embedding at the last time step, $\mathbf{e}_T$.

The method employing the LLMs-based view augmentation module generates two augmented versions of the sequence, $\tilde{S}_1$ and $\tilde{S}_2$. $Crop_p$ represents the positive sample cropping method in the previous module, and $Crop_n$ represents the negative sample cropping method. The augmentation rate is denoted as α (for example, $\alpha = 0.75$), indicating that 75% of the original sequence length is retained after cropping, resulting in the augmented sequences:

$$\tilde{S}_1 = Crop_p(S, \alpha), \quad \tilde{S}_2 = Crop_n(S, \alpha). \tag{5}$$

These two augmented sequences, $\tilde{S}_1$ and $\tilde{S}_2$, are fed into the model, and the SASRec model generates their embedding representations:

$$\tilde{\mathbf{e}}_1 = f(\tilde{S}_1), \quad \tilde{\mathbf{e}}_2 = f(\tilde{S}_2). \tag{6}$$

In the contrastive learning module, the similarity between each pair of augmented samples is calculated in the embedding space, and a contrastive loss is constructed. This ensures that the model can more effectively learn latent representation features during the optimization process by considering the distances between similar sample pairs and dissimilar sample pairs. To enhance the model's capacity to differentiate between positive and negative samples in contrastive learning, a weighted InfoNCE loss function is utilized. Given two augmented sequences $\tilde{S}_1$ and $\tilde{S}_2$, with their embedding representations $\tilde{\mathbf{e}}_1$ and $\tilde{\mathbf{e}}_2$, the similarity between these embeddings is calculated using the following formula:

$$\mathcal{L}_{\text{WCL}} = -\log \frac{\exp(\tilde{\mathbf{e}}_1 \cdot \tilde{\mathbf{e}}_2/\tau)}{\exp(\tilde{\mathbf{e}}_1 \cdot \tilde{\mathbf{e}}_2/\tau) + \sum_{i=1}^{K} w_i \exp(\tilde{\mathbf{e}}_1 \cdot \tilde{\mathbf{e}}_i^-/\tau)}, \tag{7}$$

where $\tilde{\mathbf{e}}_1$ and $\tilde{\mathbf{e}}_2$ are the embedding representations of two augmented sequences. τ is the temperature coefficient used to adjust the scale of similarity. An independent weight term $w_i \in (0, 1)$ is introduced for each negative sample to directly control its penalty strength.

Furthermore, the weighted InfoNCE loss function modifies the similarity measure according to the difficulty or significance of the sample pairs, thus improving the model's capability to differentiate between various sample pairs throughout training.

During training, the model parameters θ are refined by minimizing the overall loss function $\mathcal{L}_{\text{WCL}}$. A conventional optimization algorithm is employed, utilizing gradient descent to optimize the loss function. The optimization goal is detailed as follows:

$$\theta_{\text{opt}} = \arg\min_{\theta} \left(\lambda \mathcal{L}_{\text{WCL}}\right), \tag{8}$$

where λ is an adjustable weight hyperparameter. Theoretically, the existing SR model can be integrated with this module to enhance its performance and adapt to various recommendation scenarios.

4.3 Sequential Recommendation Module

In this module, the Linear Recurrent Units (LRU) are integrated as the foundational model. The model is designed to address recommendation problems with temporal dependencies, using a weighted loss function to achieve high performance across various recommendation tasks. We adopt the

Cross-Entropy Loss function to measure the model's prediction accuracy. In addition to the standard Cross-Entropy Loss, we introduce a weight parameter λ_s to further optimize the model's performance on specific tasks and prevent overfitting. The final combined loss function is then obtained. Its specific form is as follows:

$$\mathcal{L}_{\text{seq}} = \sum_{i=1}^{N} \left(\lambda_s \cdot \mathcal{L}_{\text{CE}}^{(i)} \right), \tag{9}$$

where N denotes the batch size, $\mathcal{L}_{\text{CE}}^{(i)}$ represents the cross-entropy loss for the i-th sample, and λ_s are hyperparameters employed to regulate the impact of the loss term on the ultimate training objective.

This sequence recommendation module, by combining the LRU with variants of the cross-entropy loss function, provides a flexible and efficient solution to sequence recommendation tasks. By adjusting the hyperparameters within the loss function, the model can be optimized according to the specific needs of the task, thereby achieving personalized recommendation results.

4.4 Training Strategy

In this section, the proposed training strategy module aims to effectively integrate the view augmentation module and the contrastive learning module with the sequential recommendation module.

The view augmentation module and the contrastive learning module are trained through unsupervised learning techniques, which removes the requirement for labeled data. During this process, the original data is augmented through the view enhancement module, and the model parameters are optimized using contrastive learning loss. The objective of this stage is to learn the latent representations of the data, enhancing the model's ability to understand the input data.

The training process combines the contrastive learning module and the sequential recommendation module, optimizing them through a weighted combination of their respective loss functions, and jointly training the model via backpropagation. This approach enables the model to simultaneously optimize the latent representations and the recommendation task, thereby learning richer feature representations.

The total training loss function, denoted as $\mathcal{L}_{\text{joint}}$, consists of two components: the contrastive learning loss $\mathcal{L}_{\text{WCL}}$ and the sequential recommendation loss $\mathcal{L}_{\text{seq}}$. The weight parameters λ_1 and λ_2 are used to adjust the contributions of the respective modules. The formula is given by:

$$\mathcal{L}_{\text{joint}} = \lambda_1 \mathcal{L}_{\text{WCL}} + \lambda_2 \mathcal{L}_{\text{seq}}. \tag{10}$$

During the training process, both modules are optimized through joint backpropagation. This strategy fully leverages the advantages of unsupervised learning by jointly optimizing the contrastive learning and SR tasks, reducing the reliance on labeled data. By introducing weight parameters λ_1 and λ_2 into the total loss function, the impact of each module can be adjusted as needed. This method enables the model to more effectively capture users' latent preferences and behavioral patterns, thereby enhancing its generalization capabilities and accuracy in practical recommendation tasks. The pseudocode for the overall algorithm is presented in Algorithm 1.

Algorithm 1: CLLMRec

Input: Input sequence data S, learning rate lr, hyperparameter λ_a, α, w_i, λ, λ_s, λ_1, λ_2, epochs E_1 (view augmentation & contrastive learning), E_2 (sequential recommendation), E_3 (joint training), batch sizes B_1, B_2, B_3.
Output: Global model parameters $\mathscr{W}$.

Stage 1: view augmentation and Contrastive Learning
for $i \leftarrow 1$ **to** E_1 **do**
  for $j \leftarrow 1$ **to** B_1 **do**
    Augment sequence data S using the LLMs to augment views $\tilde{S}_1$ and $\tilde{S}_2$;
    Perform contrastive learning with $\tilde{S}_1$ and $\tilde{S}_2$;
    Compute contrastive loss $\mathcal{L}_{\text{WCL}}$ (Eq. (7));
    Store $\mathcal{L}_{\text{WCL}}$, no parameter update;
Stage 2: Sequential Recommendation Computation
for $i \leftarrow 1$ **to** E_2 **do**
  for $j \leftarrow 1$ **to** B_2 **do**
    Compute recommendation loss $\mathcal{L}_{\text{seq}}$ (Eq. (9));
    Store $\mathcal{L}_{\text{seq}}$, no parameter update;
Stage 3: Joint Optimization
for $i \leftarrow 1$ **to** E_3 **do**
  for $j \leftarrow 1$ **to** B_3 **do**
    Compute joint loss: $\mathcal{L}_{\text{joint}} = \lambda_1 \mathcal{L}_{\text{WCL}} + \lambda_2 \mathcal{L}_{\text{seq}}$;
    Minimize $\mathcal{L}_{\text{joint}}$ using optimizer;
    Update model parameters $\mathscr{W}$;
return $\mathscr{W}$

5 Experiment

To assess the effectiveness of the proposed approach, the experimental design includes both comparative and ablation experiments, aiming to address the following three key questions:

RQ1: Does the proposed module improve the overall performance of the model?

RQ2: What are the specific contributions of each module in enhancing model performance?

RQ3: What impact do various hyperparameter configurations have on the model's performance?

5.1 Experimental Setup

Datasets. The experiments were conducted using three publicly available recommendation system datasets: ML-1M, Beauty, and Steam, which cover multiple domains including movies, cosmetics, and games, making them highly representative. The ML-1M dataset is a movie rating dataset with dense user behavior, containing a large number of user-item interaction records, making it suitable for evaluating the

Dataset	Metric	NARM	GRU4Rec	SASRec	BERT4Rec	FMLP-Rec	HSTU	LRURec	CLLMRec	Improv.
ML-1M	NDCG_{fc}@10	0.15302	0.15901	0.18199	0.16361	0.15291	0.18938	0.19044	**0.20014**	5.09 %
	NDCG_{fc}@20	0.17765	0.18694	0.21216	0.19104	0.17906	0.21521	0.21643	**0.22624**	4.53%
	Recall@10	0.27284	0.28249	0.31397	0.30873	0.29068	0.31221	0.32320	**0.33519**	3.71%
	Recall@20	0.37061	0.39326	0.43069	0.41726	0.39449	0.43002	0.43105	**0.43898**	1.84%
Beauty	NDCG_{fc}@10	0.01879	0.01969	0.02902	0.02417	0.02510	0.03008	0.03012	**0.03145**	4.42%
	NDCG_{fc}@20	0.02232	0.02327	0.03397	0.02928	0.02938	0.03527	0.03523	**0.03596**	1.96%
	Recall@10	0.03308	0.03362	0.05023	0.04671	0.04615	0.05270	0.05268	**0.05376**	2.01%
	Recall@20	0.04701	0.04724	0.06946	0.06719	0.06332	0.07137	0.07134	**0.07224**	1.22%
Steam	NDCG_{fc}@10	0.06473	0.06502	0.06734	0.06426	0.06076	0.06821	0.06837	**0.07017**	2.63%
	NDCG_{fc}@20	0.07987	0.07994	0.08286	0.08112	0.07514	0.08351	0.08390	**0.08552**	1.93%
	Recall@10	0.12153	0.12110	0.12583	0.12181	0.11982	0.12821	0.12813	**0.12998**	1.38%
	Recall@20	0.18187	0.18303	0.18705	0.18236	0.17715	0.19022	0.19018	**0.19212**	1.00%

Table 1: Comparison of evaluation metrics results: top results are highlighted in bold, with the second-best results underlined. Our proposed framework CLLMRec demonstrates superior recommendation accuracy and generalization capabilities relative to other baselines.

model's performance in high-frequency interaction scenarios. The Beauty and Steam datasets, on the other hand, have sparser interactions, providing a solid foundation for evaluating the model's performance in sparse data settings.

Evaluation metrics. To comprehensively assess the performance of the recommendation model, two main evaluation metrics were used in the experiments: NDCG@K (full corpus) and Recall@K. These two metrics measure the quality of the recommendation results from different dimensions. NDCG (Normalized Discounted Cumulative Gain) is a widely utilized metric for assessing the ranking quality in recommendation systems. In the full corpus setting, the calculation of NDCG@K (full corpus) considers the global performance across the entire dataset, meaning it averages the NDCG@K values of the recommendation results for all users. Recall is an important metric for measuring the coverage ability of a recommendation system, representing the ratio of the number of relevant items in the recommendation list to the total number of relevant items for the user.

Baselines. This model uses the LRURec and SASRec models as the base recommendation models. To evaluate the effectiveness of recommendations for users, we compare seven recommendation models across three datasets. NARM employs an RNN to predict the next item by analyzing user behaviors [Li *et al.*, 2017]. GRU4Rec utilizes gated recurrent units to predict user-item interactions [Hidasi and Karatzoglou, 2018]. SASRec uses unidirectional self-attention to capture temporal relationships between users and items [Kang and McAuley, 2018]. BERT4Rec captures complex patterns in user sequences with Bert [Sun *et al.*, 2019]. FMLP-Rec integrates an MLP with FFT for learning in complex domains [Zhou *et al.*, 2022]. HSTU enhances generative recommendations by efficiently handling high-cardinality and non-stationary streaming data. [Zhai *et al.*, 2024]. LRURec reduces computational overhead by optimizing linear recurrence operations for incremental inference [Yue *et al.*, 2024].

Implementation Details. The baseline methods in this study are implemented by referring to the original papers and the official source code. Models are trained using the cross-entropy loss function and the AdamW optimizer, with a batch size set at 128, a learning rate of 1e-3, and a maximum of 2000 training epochs. Validation occurs every 10 epochs during training. The view augmentation rate α is set at 0.75, and the contrastive learning weight w_i is set at 0.5. Early stopping is employed when Recall@10 shows no improvement for 20 consecutive validation rounds. For fairness, all baseline models are evaluated under identical hyperparameter settings and follow the design specifications outlined in their respective original studies.

5.2 Performance Comparison Analysis (RQ1)

CLLMRec was evaluated against several baseline models across three datasets, with results shown in Table 1. By comparing CLLMRec with the baseline models, the specific improvement margins were calculated. The results in the table show that CLLMRec outperforms all other baseline models in every recommendation scenario. Specifically, using the ML-1M dataset as an example, compared to the best baseline model LRURec, CLLMRec achieved a 5.09% improvement in NDCG@10, a 4.53% improvement in NDCG@20, a 3.71% improvement in Recall@10, and a 1.84% improvement in Recall@20. These results indicate that CLLMRec significantly outperforms other models in capturing subtle user preferences and enhancing recommendation accuracy.

In all experimental scenarios, CLLMRec consistently outperformed the best baseline models. This advantage can be attributed to CLLMRec's innovative design in view augmentation and contrastive learning. By leveraging LLMs for view augmentation, CLLMRec generates diverse user behavior sequences, enhancing the model's sensitivity to subtle variations in these sequences. This capability is crucial for addressing issues such as data sparsity and low-frequency behaviors. Furthermore, the integration of a contrastive learning framework allows the augmented data to assist the model in better identifying complex user behavior patterns during training, effectively reducing noise interference. This significantly improves the model's stability and accuracy. Compared to traditional methods, CLLMRec demonstrates superior efficiency in handling high-dimensional data and exhibits stronger generalization capabilities across multiple publicly available datasets. These advantages collectively enable CLLMRec to surpass the best baseline models across all evaluation metrics.

5.3 Ablation Study (RQ2)

Compared with traditional sequence recommendation models, CLLMRec integrates the LLMs-based view augmentation module and the contrastive learning module. Ablation experiments were conducted to evaluate the effectiveness of these two modules, and the following two variant models were designed for comparison:

- CLLMRec-w/o LLM: The LLMs-based view augmentation module is removed, and traditional contrastive learning methods are used. This experiment aims to verify the importance of the LLMs in the view augmentation process, especially in generating diverse user behavior sequences, improving recommendation accuracy, and handling data sparsity.
- CLLMRec-w/o CL: Further simplified, both the view augmentation and contrastive learning modules are removed, and only self-supervised learning is used for training. This experiment aims to evaluate the critical role of view augmentation and contrastive learning in improving model generalization and capturing complex user behavior patterns.

The following observations can be drawn from the data in Table 2. First, removing the LLMs module (i.e., -w/o LLM) significantly reduced the model's performance across all recommendation scenarios. This is because CLLMRec-w/o LLM relies solely on traditional view augmentation methods to generate user behavior sequences, lacking the capability of the LLMs. Consequently, the inability to effectively leverage pre-trained LLMs for optimizing the sequence generation strategy, coupled with a lack of precise adjustments based on the significance of key attributes, results in a deviation between the generated sequences and actual user preferences, thereby compromising recommendation quality.

Secondly, removing both the view augmentation and contrastive learning modules (i.e., -w/o CL) further decreased the model's performance. CLLMRec-w/o CL relies solely on self-supervised learning, which hinders the model's ability to effectively identify details and potential patterns in user behavior sequences. As the model only learns from a limited amount of raw data, it fails to capture user preferences sufficiently, resulting in reduced recommendation accuracy and limited generalization ability. The complete ablation study results are shown in Fig. 2.

5.4 Parameter Influence (RQ3)

To analyze the impact of hyperparameters on model performance, multiple sets of experiments were conducted, including adjustments to the augmentation ratio and weight coefficients in the loss function. By comparing the model's performance under different configurations, the sensitivity of hyperparameters to recommendation effectiveness was evaluated. The experimental results indicate that the augmentation ratio significantly affects model accuracy. A too-low ratio results in insufficient data diversity, making it difficult to enhance the model's ability to recognize subtle differences in user behavior. On the other hand, a ratio that is too high may introduce noise, reducing the model's stability. The weight coefficients in the loss function are equally important; appropriate coefficients can balance the roles of contrastive learning and view augmentation, thereby improving model performance. The experiments demonstrate that proper configuration of the weight coefficients effectively captures the details of user behavior sequences and enhances recommendation accuracy. Overall, this study provides clear guidance for hyperparameter optimization, and the reasonable configuration of these hyperparameters significantly improves the performance of the CLLMRec framework in both recommendation accuracy and generalization ability.

Method	ML-1M		Beauty		Steam	
	NDCG	Recall	NDCG	Recall	NDCG	Recall
-w/o LLM	0.19528	0.33002	0.03103	0.05312	0.06934	0.12908
-w/o CL	0.19044	0.32320	0.03012	0.05268	0.06837	0.12813
CLLMRec	0.20014	0.33519	0.03145	0.05376	0.07017	0.12998

Table 2: Ablation study results for CLLMRec and its variants. Both the LLM module and the CL module significantly enhance the recommendation performance.

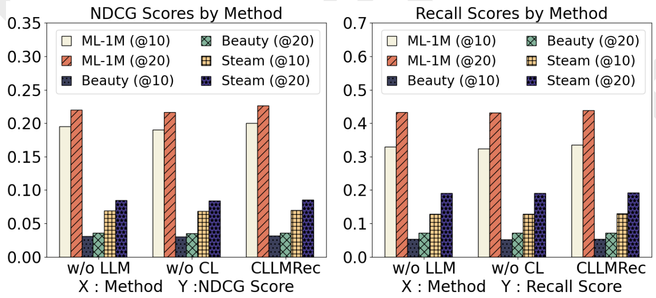

Figure 2: Comprehensive ablation study results comparison across three metrics. The enhancements from the two modules are most pronounced on the dense dataset ML-1M.

6 Conclusion

This paper presents the CLLMRec framework, which combines LLMs for view augmentation and introduces a contrastive learning framework. By leveraging augmented data for effective contrastive learning, it addresses the issues of data sparsity and the neglect of irrelevant noise commonly encountered in SR tasks. Extensive experiments were conducted on three publicly available datasets, demonstrating that CLLMRec outperforms existing state-of-the-art baseline models in diverse SR scenarios. Additionally, further ablation experiments thoroughly confirmed the robustness and effectiveness of this framework in improving the performance of SR models across various practical settings.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 92267104 and Jiangsu Provincial Major Project on Basic Research of Cutting-edge and Leading Technologies, under grant no. BK20232032.

References

[Chen *et al.*, 2024] Wangmei Chen, Zepeng Shen, Yiming Pan, Kai Tan, and Cankun Wang. Applying machine learning algorithm to optimize personalized education recommendation system. *Journal of Theory and Practice of Engineering Science*, 4(01):101–108, 2024.

[Hao *et al.*, 2023] Yongjing Hao, Pengpeng Zhao, Xuefeng Xian, Guanfeng Liu, Lei Zhao, Yanchi Liu, Victor S Sheng, and Xiaofang Zhou. Learnable model augmentation contrastive learning for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[He *et al.*, 2023] Wei He, Guohao Sun, Jinhu Lu, and Xiu Susie Fang. Candidate-aware graph contrastive learning for recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1670–1679, 2023.

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

[Jiang *et al.*, 2023] Yangqin Jiang, Chao Huang, and Lianghao Huang. Adaptive graph contrastive learning for recommendation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 4252–4261, 2023.

[Jiang *et al.*, 2024] Meng Jiang, Keqin Bao, Jizhi Zhang, Wenjie Wang, Zhengyi Yang, Fuli Feng, and Xiangnan He. Item-side fairness of large language model-based recommendation system. In *Proceedings of the ACM on Web Conference 2024*, pages 4717–4726, 2024.

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

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

[Li *et al.*, 2024] Nian Li, Xin Ban, Cheng Ling, Chen Gao, Lantao Hu, Peng Jiang, Kun Gai, Yong Li, and Qingmin Liao. Modeling user fatigue for sequential recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 996–1005, 2024.

[Liang *et al.*, 2024] Mengkun Liang, Liping Yang, Kai Li, and Hao Zhai. Improved collaborative filtering for cross-store demand forecasting. *Computers & Industrial Engineering*, 190:110067, 2024.

[Liao *et al.*, 2024] 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.

[Lin *et al.*, 2024] Xinyu Lin, Wenjie Wang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Bridging items and language: A transition paradigm for large language model-based recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1816–1826, 2024.

[Qin *et al.*, 2024] Yifang Qin, Wei Ju, Hongjun Wu, Xiao Luo, and Ming Zhang. Learning graph ode for continuous-time sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[Ren *et al.*, 2024] 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 on Web Conference 2024*, pages 3464–3475, 2024.

[Sun *et al.*, 2019] 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.

[Xu *et al.*, 2024] Xiaolong Xu, Hongsheng Dong, Lianyong Qi, Xuyun Zhang, Haolong Xiang, Xiaoyu Xia, Yanwei Xu, and Wanchun Dou. Cmclrec: Cross-modal contrastive learning for user cold-start sequential recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1589–1598, 2024.

[Xu *et al.*, 2025] Xiaolong Xu, Hongsheng Dong, Haolong Xiang, Xiyuan Hu, Xiaoyong Li, Xiaoyu Xia, Xuyun Zhang, Lianyong Qi, and Wanchun Dou. C2lrec: Causal contrastive learning for user cold-start recommendation with social variable. *ACM Transactions on Information Systems*, 2025.

[Yang *et al.*, 2023] Yonghui Yang, Zhengwei Wu, Le Wu, Kun Zhang, Richang Hong, Zhiqiang Zhang, Jun Zhou, and Meng Wang. Generative-contrastive graph learning for recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1117–1126, 2023.

[Yang *et al.*, 2024a] 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.

[Yang *et al.*, 2024b] Zhengyi Yang, Jiancan Wu, Zhicai Wang, Xiang Wang, Yancheng Yuan, and Xiangnan He. Generate what you prefer: Reshaping sequential recommendation via guided diffusion. *Advances in Neural Information Processing Systems*, 36, 2024.

[Yue *et al.*, 2024] Zhenrui Yue, Yueqi Wang, Zhankui He, Huimin Zeng, Julian McAuley, and Dong Wang. Linear recurrent units for sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 930–938, 2024.

[Zhai *et al.*, 2024] Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Jiayuan He, et al. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. In *Forty-first International Conference on Machine Learning*, 2024.

[Zhang *et al.*, 2023] Junjie Zhang, Ruobing Xie, Yupeng Hou, Xin Zhao, Leyu Lin, and Ji-Rong Wen. Recommendation as instruction following: A large language model empowered recommendation approach. *ACM Transactions on Information Systems*, 2023.

[Zhang *et al.*, 2024a] An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. On generative agents in recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in Information Retrieval*, pages 1807–1817, 2024.

[Zhang *et al.*, 2024b] Yabin Zhang, Zhenlei Wang, Wenhui Yu, Lantao Hu, Peng Jiang, Kun Gai, and Xu Chen. Soft contrastive sequential recommendation. *ACM Transactions on Information Systems*, 2024.

[Zhang *et al.*, 2024c] Ye Zhang, Yanqi Gao, Yupeng Zhou, Jianan Wang, and Minghao Yin. Mrmlrec: A two-stage approach for addressing data sparsity in mooc video recommendation (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23709–23711, 2024.

[Zhao *et al.*, 2024] 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). *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[Zheng *et al.*, 2024] 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*, pages 1435–1448. IEEE, 2024.

[Zhou *et al.*, 2022] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. Filter-enhanced mlp is all you need for sequential recommendation. In *Proceedings of the ACM web conference 2022*, pages 2388–2399, 2022.