MCL4SRec: A Sequential Recommendation Model with Multi-level Contrastive Learning

Zhuohan Hu
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, China
huzhuohan2020@gmail.com

Bo Yang*
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, China
ORCID: 0000-0003-0805-7928

Jialiang Lin
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, China
lin1042467351@gmail.com

Jiajin Wu
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, China
wujiajin@std.uestc.edu.cn

Wei Liu
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, China
liu_wei@std.uestc.edu.cn

***Abstract*—Sequential recommendation (SR) plays an important role across various platforms, aiming to predict users' next items of interest based on their historical interaction sequences. Recent SR studies have employed deep learning techniques, such as Recurrent Neural Networks and Self-Attention (SA) mechanism, demonstrating promising results. Inspired by the emergence of contrastive learning methods, some SR models have utilized contrastive learning to improve the accuracy of recommendations. However, existing SR models employing contrastive learning primarily construct positive and negative sample pairs only from user interaction sequences, i.e., through *sequence-level* contrastive learning. In our research, we argue that there also exists semantic similarities between items, which can be used to conduct the *item-level* constructive learning, resulting in better recommendation accuracy. In this paper, we propose MCL4SRec, an SA-based SR model that combines sequence-level and item-level contrastive learning to enhance recommendation accuracy. In our proposed MCL4SRec, the item-level contrastive learning module utilizes items' category information to construct positive and negative sample pairs, capturing semantic similarities and differences between items. Additionally, in MCL4SRec, we propose to use more side information such as category and brand to further improve the accuracy of recommendations. We conduct extensive experiments on three widely-used datasets to evaluate the proposed MCL4SRec. Experimental results indicate that the average improvements compared with the recent well-known baselines range from 7.73% to 16.18% in HR and NDCG, demonstrating the effectiveness of MCL4SRec for SR tasks.**

***Index Terms*—sequential recommendation, contrastive learning, item-level, sequence-level, self-attention mechanism**

*Bo Yang is the corresponding author.

I. Introduction

Recommender systems are now widely applied across various online platforms (e.g., TikTok, Taobao, and Amazon) to help users find information that interests them. In real-world scenarios, users' preferences are often dynamic and evolving over time, making it a challenge to make appropriate recommendations. Therefore, sequential recommendation (SR) [1] [2] [3] [4] [5] [6] [7] has been proposed and attracted much attention in both academia and industry recently. The goal of SR is to predict the items that users may interact with by modeling the temporal dependencies in the user interaction sequences [1].

Various types of SR models have been proposed. Early SR models adopted Markov chains [8] [9]. With the development of deep learning techniques, many SR models employ deep neural networks to model users' dynamic preferences, such as Recurrent Neural Network (RNN) [10] [11] and Convolutional Neural Network (CNN) [12] [13]. More recently, Self-Attention (SA) mechanism has also been adopted by SR models [2] [3] [4]. Since the SA mechanism can simultaneously capture the long-term and short-term dependencies of user interaction sequences [2], recent SR models based on self-attention mechanism have achieved state-of-the-art (SOTA) performance.

In recent years, contrastive learning has been introduced into SR to improve the accuracy of SR models [14] [15] [16] [17]. The contrastive learning aims to bring positive samples closer in the feature space and push negative samples further apart. Existing SR models with contrastive learning primarily

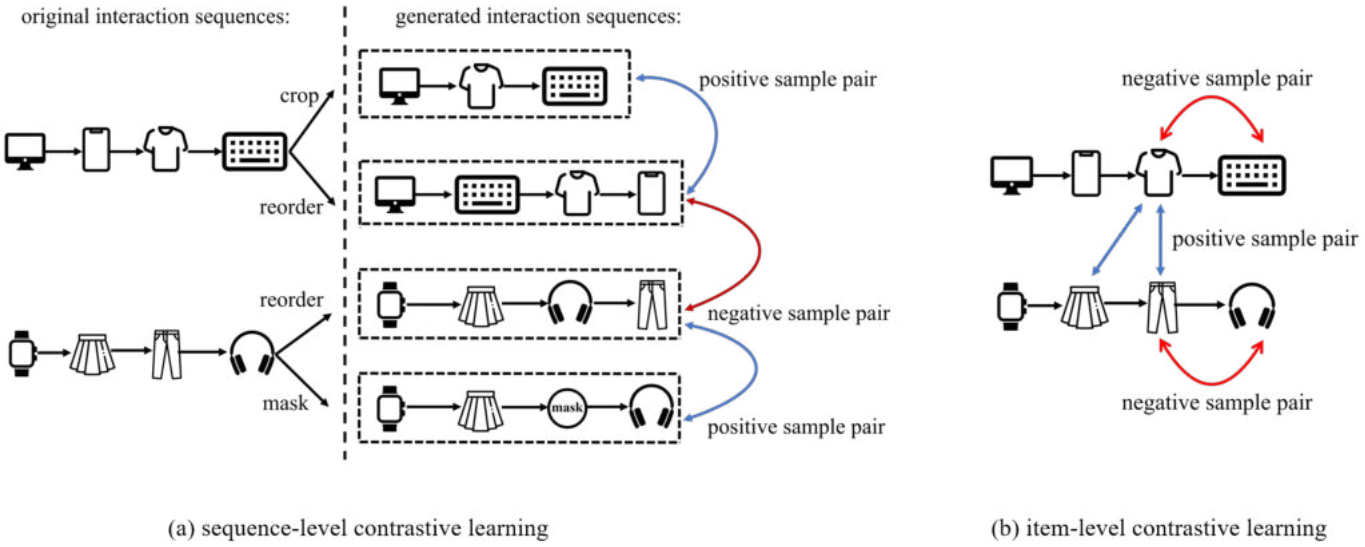

Fig. 1. Two different types of contrastive learning methods in SR.

construct positive and negative sample pairs only from user interaction sequences, i.e., through *sequence-level* contrastive learning [15] [16] [17]. Fig. 1(a) provides an example to illustrate sequence-level contrastive learning. The sequences on the left are the users' original interaction sequences, and the sequences on the right are generated through data augmentation approaches (e.g., crop, reorder, mask). If the generated sequences come from the same sequence, then they are regarded as a positive sample pair (e.g., sequences I and II, III and IV); otherwise, they are regarded as a negative sample pair (e.g., sequence II and III).

In our research, we argue that besides sequence-level contrastive learning, *item-level* contrastive learning should also be introduced. Fig. 1(b) shows a simple example to illustrate item-level contrastive learning. In the users' original interaction sequences, semantically similar items such as "T-shirt" and "skirt" can be regarded as a positive sample pair since both are in the category of "clothes", while dissimilar items like "T-shirt" and "keyboard" can be regarded as a negative sample pair. By this approach, the semantic similarity and difference of items can be captured and utilized in the contrastive learning, which could lead to an improvement of the accuracy of an SR model.

Obviously, for above-mentioned item-level contrastive learning, the key is how to determine which items are positive sample pairs and which are negative sample pairs. To achieve this, we propose to utilize the items' category information to help identify positive and negative sample pairs. It can be noted that category information is generally available and has been used in some SR and session-based recommendation models [17] [18] [19].

In this paper, we propose a novel SA-based SR model, named **M**ulti-level **C**ontrastive **L**earning for **S**equential **R**ecommendation (MCL4SRec). In MCL4SRec, there are mainly four key components: (1) A user interaction sequence encoder combining embedding representations of items, categories, and brands. Specifically, for each item in the user interaction sequence, we propose to learn three different embedding representations: one for the item itself, one for the item's category, and one for the item's brand. (2) Item-level contrastive learning module. We propose to use items' category information to group items and calculate the *central embedding* in each group. The obtained central embedding can be used to construct positive and negative sample pairs at the item level, which will be introduced in detail in III-C. (3) Sequence-level contrastive learning module. We also employ three different data augmentation approaches to construct positive and negative sample pairs at the sequence level, including crop, mask, and reorder, which are widely used in existing SR models [15] [16] [17]. (4) A multi-task training strategy. MCL4SRec adopts a multi-task training strategy to jointly optimize the contrastive learning task and SR task.

The main contributions of this paper can be summarized as follows:

- To the best of our knowledge, this is the first work to introduce item-level contrastive learning in SR.
- We propose a method of constructing positive and negative sample pairs in item-level contrastive learning.
- We propose a novel SR Model MCL4SRec based on contrastive learning, which combines sequence-level contrastive learning with item-level contrastive learning, thereby enhancing the accuracy of recommendations.

II. Preliminaries

A. Notations

We denote the user and item sets as $U = \left[u_1, u_2, \ldots, u_{|U|}\right]$ and $I = \left[i_1, i_2, \ldots, i_{|I|}\right]$. Likewise, the sets of all categories and brands can be denoted as $C = \left[c_1, c_2, \ldots, c_{|C|}\right]$ and $B = \left[b_1, b_2, \ldots, b_{|B|}\right]$. Each item $i_t \in I$ belongs to a

certain category $c_j \in C$ and a certain brand $b_k \in B$, and a category or brand may contain many items. For each user $u \in U$, we can depict a sequence of items they have interacted with as a chronologically ordered list $V^u = [v_1^u, v_2^u, \ldots, v_l^u]$. And the corresponding category sequence and brand sequence can be denoted as $C^u = \left[c_{v_1}^u, c_{v_2}^u, \ldots, c_{v_l}^u\right]$ and $B^u = \left[b_{v_1}^u, b_{v_2}^u, \ldots, b_{v_l}^u\right]$, respectively. Here, l indicates the count of items in V^u, and v_t^u is the tth item interacted in the current sequence. The embedding representation of V^u can be denoted as $S^u = [S_1^u, S_2^u, \ldots, S_l^u]$, where S_t^u represents the embedding of the tth interacted item. And we use h^u to represent the final embedding representation of user u.

B. Task Definition

The task of SR involves predicting the item that user u is most likely to interact with at time step $t+1$, based on the sequence V^u, which consists of the T most recent interacted items at time step t. For SR task, we can use the BPR pairwise ranking loss to train the model:

$$\mathcal{L}_{Rec} = \sum_{(h^u, S_t)} -\log \sigma(\hat{y}(h^u, S_t) - \hat{y}(h^u, S_t^-)), \tag{1}$$

where S_t^- is the embedding of a randomly sampled negative item i_t^- that user does not interact with and σ is the sigmoid function.

III. METHODOLOGY

In this section, we introduce our proposed model, the Multi-level Contrastive Learning for Sequential Recommendation (MCL4SRec). The framework of MCL4SRec is illustrated in Fig. 2, which consists of four main components: a user interaction sequence encoder, a sequence-level contrastive learning module, an item-level contrastive learning module, and a multi-task learning framework.

A. User Interaction Sequence Encoder

In this subsection, we will introduce how MCL4SRec obtains the embedding representations of items and users. For the input item sequence $V^u = [v_1^u, v_2^u, \ldots, v_l^u]$, we utilize a trainable item ID embedding matrix $M_{id} \in \mathbb{R}^{N \times d}$ and perform a look-up operation to extract the embedding vector for each item $v_t^u \in V^u$, which can be denoted as $e_{v_t}^u \in \mathbb{R}^{1 \times d}$. Then the embedding representation of the item sequence of user u can be expressed as:

$$E_{id}^u = \left[e_{v_1}^u, e_{v_1}^u, \ldots, e_{v_l}^u\right] \tag{2}$$

Similarly, for the corresponding items' category sequence $C^u = \left[c_{v_1}^u, c_{v_2}^u, \ldots, c_{v_n}^u\right]$ and items' brand sequence $B^u = \left[b_{v_1}^u, b_{v_2}^u, \ldots, b_{v_n}^u\right]$, we propose to introduce two additional trainable embedding matrices $M_{ca} \in \mathbb{R}^{I \times d}$, $M_{br} \in \mathbb{R}^{M \times d}$. Utilizing the look-up operation again, we obtain the embedding representations for both sequences, denoted as E_{ca}^u and E_{br}^u respectively:

$$E_{ca}^u = \left[e_{c_{v_1}}^u, e_{c_{v_2}}^u, \ldots, e_{c_{v_l}}^u\right] \tag{3}$$

$$E_{br}^u = \left[e_{b_{v_1}}^u, e_{b_{v_2}}^u, \ldots, e_{b_{v_l}}^u\right] \tag{4}$$

To enrich the semantic information and capture user preferences more effectively, we propose concatenating the embedding representations at corresponding positions of the three embedding representations. This results in a more comprehensive embedding representation $E^u \in \mathbb{R}^{L \times 3d}$:

$$E^u = \left[e_1^u, e_2^u, \ldots, e_l^u\right], \tag{5}$$

where $e_t^u = e_{v_t}^u \parallel e_{c_{v_t}}^u \parallel e_{b_{v_t}}^u$ is the embedding vector of item v_t^u which has integrated the item's category and brand information. The $\parallel$ symbol represents the concatenation operation. This approach could effectively captures user preferences, as both the category and brand of an item influence a user's preference to some extent.

Furthermore, to preserve the temporal information within the sequence, we leverage a learnable positional embedding $P \in \mathbb{R}^{L \times 3d}$, where L is the maximum length of the sequence. Finally, we combine the item embedding representation E^u and the positional embedding representation P to obtain the final user representation S^u:

$$S^u = \left[s_1^u, s_2^u, \ldots, s_T^u\right], \tag{6}$$

$$s_t^u = e_t^u + p_t, \tag{7}$$

here, s_t^u is the final embedding vector of item interacted at time step t, and p_t is the positional embedding at time step t.

Following previous works [2] [14] [15], after obtaining the user embedding representation S^u, we stack the multi-head self-attention layers, denoted as $\mathrm{Trm}^L(\cdot)$ for L-layers, to aggregate sequential features, yielding:

$$H^u = \mathrm{Trm}^L\left(S^u\right). \tag{8}$$

At each time step t, the updated user representation H^u complies the features of items interacted with prior to this time step. Given that the recommendation task aims to predict items for each user u at time step $t+1$, we designate the last representation of H^u as the user's preference representation for the next moment:

$$h^u = H^u\left[-1\right]. \tag{9}$$

B. Sequence-Level Contrastive Learning Module

The key to sequence-level contrastive learning lies in the use of various data augmentation methods. Following the previous works [15] [16] [17], we also employ the same data augmentation methods, such as Crop, Mask, and Reorder. For each user interaction sequence V^u, we randomly choose two methods to obtain two generated sequences V_i^u and V_j^u. We will apply the data augmentation strategy to all interaction sequences in a batch of size N, acquiring a total of 2N generated sequences. Subsequently, these generated sequences would be encoded by the user interaction sequence encoder and obtain their embedding representations $\left[h_{a_i}^{u_1}, h_{a_j}^{u_1}, h_{a_i}^{u_2}, h_{a_j}^{u_2}, \ldots, h_{a_i}^{u_N}, h_{a_j}^{u_N}\right]$. For

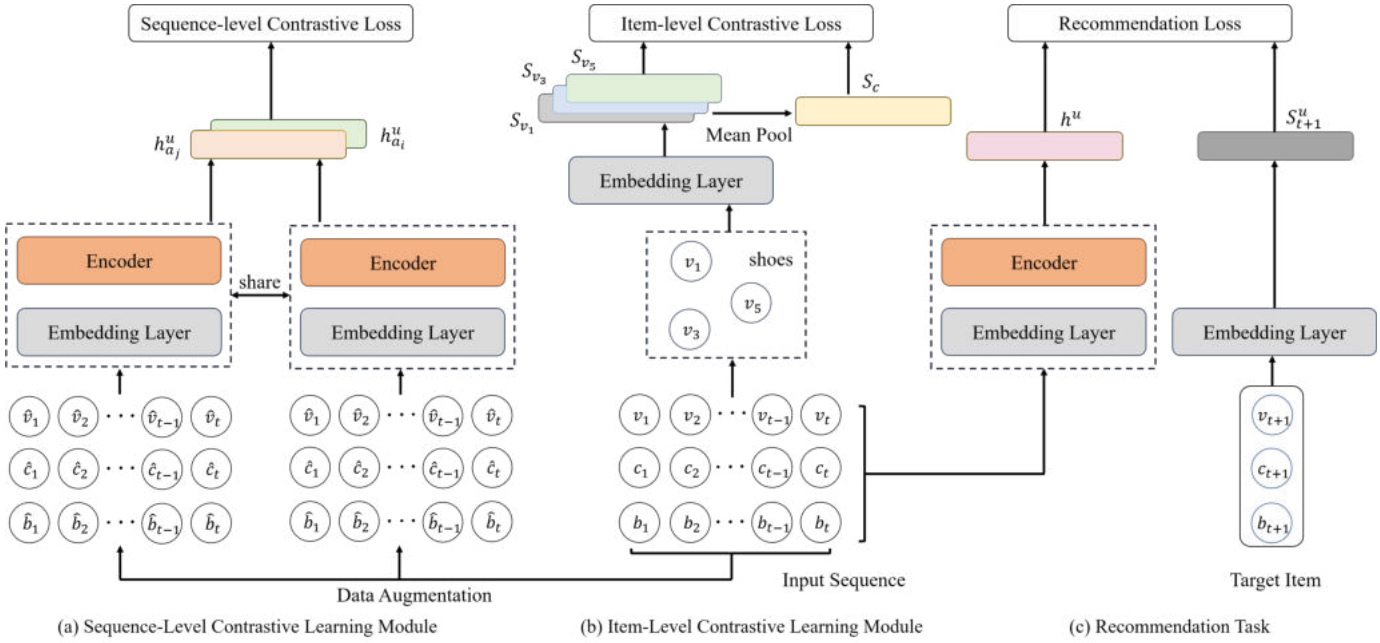

Fig. 2. The overall structure of the proposed MCL4SRec.

each user u, we regard $\left(h_{a_i}^u, h_{a_j}^u\right)$ as a positive sample pair and the other $2\,(N-1)$ samples in the same batch as negative sample pairs h^{neg}. The sequence-level contrastive learning loss for the positive sample pair $\left(h_{a_i}^u, h_{a_j}^u\right)$ can be defined as:

$$\mathcal{L}_{Seq} = -\log \frac{\exp(\text{sim}(h_{a_i}^u, h_{a_j}^u))}{\sum_{h^- \in h^{neg}} \exp(\text{sim}(h_{a_i}^u, h^-))}, \tag{10}$$

where $\text{sim}(\cdot)$ represents cosine similarity to measure similarity between two augmented samples.

C. Item-Level Contrastive Learning Module

Most existing SR models mainly focus on sequence-level contrastive learning to construct positive and negative sample pairs, which may not fully exploit the semantic similarity between items, making it difficult to learn rich item embedding representations. Therefore, we propose item-level contrastive learning for the first time and utilize item category information to help construct positive and negative sample pairs. In this subsection, we will introduce how to construct positive and negative sample pairs between items, as well as how to calculate the item-level contrastive loss.

1) Construction of positive and negative sample pairs: For the input item sequences, we first remove duplicate items from these sequences to avoid calculating loss multiple times for the same items. Then we utilize these items' category information to group them, for example, item v_1, v_3, v_5 would be assigned to the category group labeled "shoes". After that, we feed these items to the embedding layer to obtain each item embedding representation S_{v_i}.

Next, in order to construct positive and negative sample pairs for each item, we propose to calculate the central embedding representation of all items in each category, which can be expressed as:

$$S_c = \text{Avg}(\sum_i S_{v_i}^c), \tag{11}$$

where the symbol c represents these items belong to category "c", and Avg is the mean pool operation. We choose to use the mean pool operation to calculate the central embedding because it avoids introducing additional parameters, thereby reducing computational overhead. Additionally, experimental results indicate the effectiveness of this approach. For each item v_i, we treat $\left(S_{v_i}^c, S_c\right)$ as positive sample pair and consider the central embedding representations S^{neg} of the other categories as negative sample pairs.

2) Contrastive learning loss with instance weighting: Items belonging to different categories may still exhibit some semantic similarity, for example, items categorized as tablets and items categorized as smartphones. However, the model may consider them as negative sample pairs, thereby potentially reducing their semantic correlations. To avoid this problem, we employ a method of instance weighting to penalize false negatives. Consider a batch of item sequences, for each item's embedding representation S_{v_i}, and the central embedding representation $S^- \in S^{neg}$, the weight can be produced as:

$$a_{S^-} = \begin{cases} 0, & \text{sim}(S_{v_i}, S^-) \geq \phi \\ 1, & \text{sim}(S_{v_i}, S^-) < \phi \end{cases} \tag{12}$$

where ϕ is an instance weighting threshold hyper-parameter and $Sim(S_{v_i}, S^-)$ is the similarity score. Thus, the negative with the highest semantic resemblance to the central embedding representations of other categories will be considered a false negative and will be penalized with a weight of 0.

Based on the weights, we optimize the item representations with a debiased item-level contrastive learning loss function as:

$$\mathcal{L}_{Item} = -\log \frac{\exp(\text{sim}(S_{v_i}, S_c)/\tau)}{\sum_{S^- \in S^{neg}} a_{S^-} \times \exp(\text{sim}(S_{v_i}, S^-)/\tau)}, \tag{13}$$

where τ is a hyper-parameter representing the temperature, S_c is the positive sample of S_{v_i} and S^{neg} are the negative samples.

D. Multi-Task Training

The proposed MCL4SRec model would be trained with a multi-task training strategy to jointly optimize the main SR task via $\mathcal{L}_{\text{Rec}}$, the item-level contrastive learning task via $\mathcal{L}_{\text{Item}}$ and the sequence-level contrastive learning task via $\mathcal{L}_{\text{Seq}}$. Formally, we jointly train the SR model as follows:

$$\mathcal{L} = \mathcal{L}_{\text{Rec}} + \lambda_1 * \mathcal{L}_{\text{Item}} + \lambda_2 * \mathcal{L}_{\text{Seq}} \tag{14}$$

where λ_1 and λ_2 control the strengths of the item-level contrastive learning task and sequence-level contrastive learning task respectively.

IV. EXPERIMENTS

A. Datasets

The performance of MCL4SRec is evaluated using three real-world datasets(Beauty, Clothing and sports), which are from Amazon review datasets[1] and widely utilized in SR [14] [15] [16]. Following prior studies [15] [16], we interpret the existence of a review or rating as implicit feedback. Each dataset is organized by users and the item sequences are arranged in chronological order. To filter out cold-start users and items, we adhere to the standard practice of filtering out items and users with fewer than 5 feedbacks [3] [4] [15]. The statistical characteristics of these refined datasets are presented in Table. I.

TABLE I
DATA STATISTICS

Datasets	Beauty	Sports	Clothing
# Users	22,363	35,598	39,387
# Items	12,101	18,357	23,033
# categories	248	1,443	1,241
# brands	2,077	2,412	1,183
# Avg.Length	8.88	8.32	7.16
# Actions	198,502	296,337	296,337
Sparsity	99.93%	99.95%	99.95%

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

B. Baselines

In order to evaluate the effectiveness of our proposed MCL4Rec, we compare MCL4Rec with a variety of models. This includes general recommendation model that do not consider sequential dynamics (e.g., BPR), conventional sequential models (e.g., GRU4Rec, SASRec, Bert4Rec), and sequential models that incorporate contrastive learning (e.g., CL4SRec, ICLRec, MoCo4SRec).

- **BPR** [20]. It is a representative matrix factorization model that incorporates a pairwise Bayesian Personalized Ranking (BPR) loss.
- **GRU4Rec** [10]. This is an RNN-based methodology that utilizes GRU modules to represent user sequences for ranking-loss-based recommendations during a session. It is further enhanced by a novel class of loss functions and sampling technique.
- **SASRec** [1]. This is a benchmark model for addressing the SR problem. It employs a self-attention mechanism to simulate user sequences, thereby identifying the dynamic interests of users.
- **Bert4Rec** [21]. This model enhances SASRec by incorporating bidirectional self-attention modules, thereby making it a leading SR model.
- **CL4SRec** [14]. This model integrates contrastive learning with a SR model. Notably, it exclusively employs random augmentation methods for contrastive learning.
- **ICLRec** [15]. This model enhances recommendation systems through a combination of clustering and contrastive learning applied to user intentions.
- **MoCo4SRec** [16]. It proposes a Momentum contrast module and augments data in the embedding space to improve user representation and alleviate the issues of data sparsity and false negatives.

C. Experimental Settings

Consistent with prior studies [14] [15] [16] [21], we maintain an embedding size of 64 and a batch size of 256 across all models. The hyperparameters ϕ, λ_1, λ_2 are tuned within the ranges [0.1, 0.9], [0, 1], and [0.01, 0.9] respectively. After 40 epochs on the validation set, in the absence of performance improvement, we employ early stopping and report results on the test set. We utilize the Adam optimizer [22] with a learning rate lr of 0.001, β_1 of 0.9, and β_2 of 0.999 for fine-tuning the model. Our implementation is carried out using PyTorch 1.12.1 and Python 3.8. All experiments are conducted on a single NVIDIA GeForce RTX 2080Ti.

Following previous works [15] [16], we adopt the Hit Ratio@k (**HR@k**) and Normalized Discounted Cumulative Gain@k (**NDCG@k**) metrics to measure the performance. For values of k such as 5, 10 and 20, we report both the HR and NDCG metrics. The key difference between HR@k and NDCG@k lies in their focus: HR@k checks whether the target item is included in the top-k list of recommendations, while the NDCG@k considers where the target item ranks within that list.

TABLE II
PERFORMANCE COMPARISON ON THE THREE DATASETS

Dataset	Metric	BPR	GRU4Rec	SASRec	Bert4Rec	CL4SRec	ICLRec	MoCo4SRec	MCL4SRec	Impro.
Beauty	HR@5	0.0212	0.0111	0.0374	0.0351	0.0401	0.0493	0.0518	**0.0567**	9.46%
	HR@10	0.0372	0.0162	0.0575	0.0601	0.0642	0.0751	0.0756	**0.0842**	11.38%
	HR@20	0.0589	0.0478	0.0901	0.0942	0.0974	0.1076	0.1056	**0.1205**	11.98%
	NDCG@5	0.0130	0.0058	0.0241	0.0219	0.0268	0.0324	0.0346	**0.0376**	8.67%
	NDCG@10	0.0181	0.0075	0.0305	0.0300	0.0345	0.0401	0.0422	**0.0464**	9.95%
	NDCG@20	0.0236	0.0104	0.0387	0.0386	0.0428	0.0489	0.0496	**0.0556**	12.09%
Sports	HR@5	0.0141	0.0162	0.0206	0.0217	0.0231	0.0283	0.0287	**0.0316**	10.10%
	HR@10	0.0216	0.0258	0.0320	0.0359	0.0369	0.0429	0.0434	**0.0474**	9.21%
	HR@20	0.0323	0.0421	0.0497	0.0604	0.0557	0.0638	0.0640	**0.0719**	12.34%
	NDCG@5	0.0091	0.0103	0.0135	0.0143	0.0146	0.0182	0.0194	**0.0209**	7.73%
	NDCG@10	0.0115	0.0142	0.0172	0.0190	0.0191	0.0236	0.0241	**0.026**	7.88%
	NDCG@20	0.0142	0.0186	0.0216	0.0251	0.0238	0.0284	0.0293	**0.0321**	9.55%
Clothing	HR@5	0.0067	0.0095	0.0168	0.0125	0.0168	0.0173	0.0166	**0.0201**	16.18%
	HR@10	0.0094	0.0165	0.0272	0.0208	0.0266	0.0271	0.0261	**0.0298**	9.96%
	HR@20	0.0109	0.0187	0.0303	0.0235	0.0298	0.0409	0.0388	**0.0457**	11.73%
	NDCG@5	0.0052	0.0061	0.0091	0.0075	0.0090	0.0112	0.0109	**0.0123**	9.82%
	NDCG@10	0.0069	0.0083	0.0124	0.0102	0.0121	0.0143	0.014	**0.0162**	13.28%
	NDCG@20	0.0082	0.0096	0.0142	0.0123	0.0139	0.0178	0.0172	**0.0206**	15.73%

D. Overall Comparison with Baselines

Table. II displays the performance metrics of various models across three datasets, with our proposed MCL4SRec emerging as the top performer in both HR and NDCG. The top score is highlighted in bold within each row, and the second-best is indicated with underlining. The last column displays the relative improvements compared to the best baseline results. The findings derived from the table are as follows:

- A noticeable performance divide exists between sequential and non-sequential approaches. BPR consistently lags behind sequential models, underscoring the importance of extracting sequential patterns from user interaction sequences.
- Among common SR algorithms, Bert4Rec stands out, showcasing better performance attributed to its Transformer structure. This demonstrates the effectiveness of the SA mechanism in capturing user sequential information compared to CNN and RNN.
- SR models that incorporate contrastive learning consistently outperform traditional SR models in terms of performance, indicating that contrastive learning is effective in improving recommendation accuracy.
- Our proposed MCL4SRec outperforms baseline models across three datasets. For instance, MCL4SRec achieves a 12% improvement in NDCG@20 for Beauty and a 16% enhancement in HR@5 for Clothing over the best baseline models. We attribute this improvement to two factors: (1) Introducing additional item-level contrastive learning enables MCL4SRec to better capture intrinsic correlations between items, enhancing item embedding representations. (2) MCL4SRec efficiently leverages side information (e.g., category and brand) to capture user interests, thereby boosting overall performance.

E. Ablation Study

We perform ablation experiments to validate the effectiveness of item-level contrastive learning module and side information(e.g., category and brand) in improving recommendation performance. Specially, we design three contrast models:

- **MCL4SRec w/o item-level**: a variant of MCL4SRec that excludes the item-level contrastive learning module.
- **MCL4SRec w/o category**: a variant of MCL4Rec that does not utilize item category information.
- **MCL4SRec w/o brand**: a variant of MCL4SRec that does not utilize item brand information.

The results of our ablation experiments are illustrated in Fig. 3. Notably, MCL4SRec, incorporating item-level contrastive learning to capture and exploit item correlations, outperforms its counterpart without this module. This highlights the positive impact of integrating item-level contrastive learning on the model's recommendation performance. Moreover, when devoid of any side information, the model exhibits notably poor performance. However, the inclusion of either category or brand boosts the model's performance. Interestingly, our proposed model, MCL4SRec, leveraging both category and brand information, attains the highest performance. This underscores the advantage of utilizing side information associated with items to more effectively grasp users' interests and intentions. In summary, our ablation studies affirm the efficacy of item-level contrastive learning and the utilization of items' side information in enhancing recommendation performance.

V. RELATED WORK

A. Sequential Recommendation

The key idea of SR is to recommend items to users by modeling their historical interaction sequences [1] [2] [3] [4], typically utilizing RNN [10] [11] or SA mechanism [1] [2] [3] as sequence encoders. GRU4Rec [10] is the first work to utilize RNN with gated recurrent units in SR. Due to the great ability of SA mechanism, some related models have also been proposed. SASRec [1] first applies SA mechanism to assign weights to each interacted item adaptively. BERT4Rec

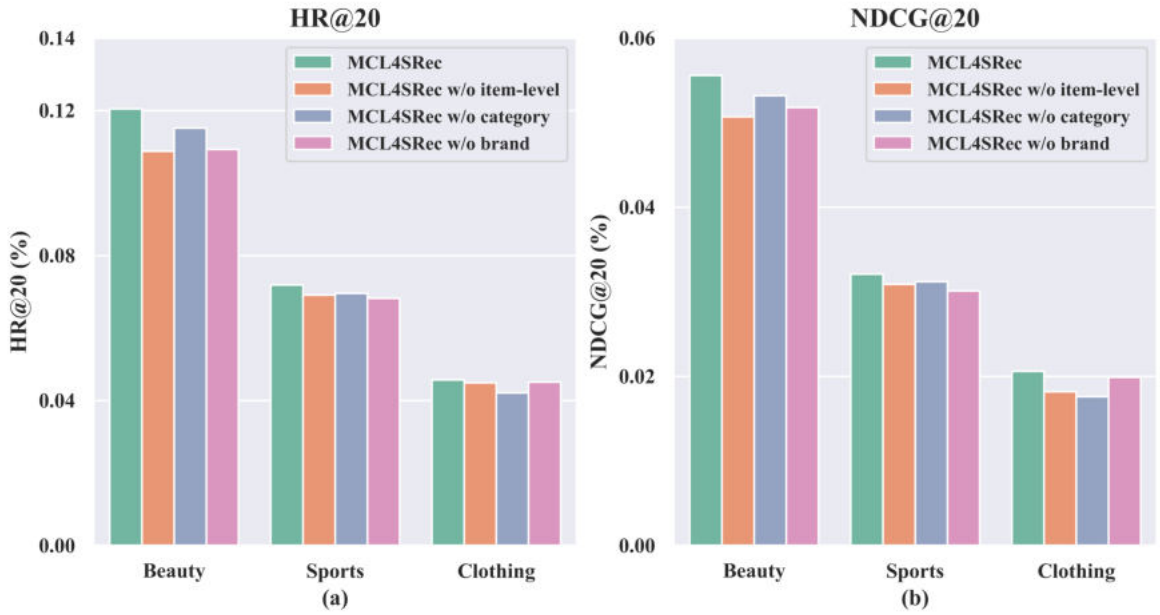

Fig. 3. Ablation Study.

[21] utilizes a bidirectional encoder to fuse user behaviors information from left and right directions into each item. TAT4SRec [2] introduces additional temporal information to better capture users' preferences. CLSR [4] is a framework based on SA mechanism which disentangles users' long and short-term interests for better recommendations. In this paper, we improve the recommendation accuracy by encoding items' side information, such as category and brand.

B. Contrastive Learning

Contrastive learning has been widely applied in various research areas, such as CV and NLP [23] [24] [25] for its strong capabilities. The fundamental idea of contrastive learning is to maximize the differences between positive and negative samples, thereby enhancing the model's feature learning and discrimination abilities for the target task. Recently, contrastive learning is also used in recommendation tasks [14] [15] [16] [17]. For the collaborative filtering methods, SGL [26] applies the NCE in node-level representation learning. SSL [27] proposes a siamese network to encode the items as pre-training with embedding-level augmentations. For the contrastive learning in SR, CL4SRec [14] employs three random augmentation techniques to construct positive and negative sample pairs. ICLRec [15] improves the accuracy of recommendations through a combination of clustering and contrastive learning applied to user intentions. Extending the CL4SRec model, CLF4SRec [28] introduces a novel frequency-domain data augmentation module to construct positive and negative sample pairs. MoCo4SRec [16] uses a momentum contrast module and augments data in the embedding space to improve user representation. Unlike these models that solely rely on sequence-level contrastive learning, our proposed model introduces additional item-level contrastive learning, achieving state-of-the-art results.

VI. CONCLUSION

In this paper, we propose a model named MCL4SRec, which combines sequence-level contrastive learning with item-level contrastive learning to better capture user preferences. Item-level contrastive learning utilizes category information of items to construct positive and negative sample pairs, enabling the capture of semantic similarity between items. Furthermore, by encoding side information of items, such as category and brand, we can further enhance recommendation accuracy. The experiments on three real-world datasets demonstrate the effectiveness of our proposed MCL4SRec.

ACKNOWLEDGMENT

This work is supported by Natural Science Foundation of Sichuan Province (Project No. 2024NSFSC0502).

REFERENCES

[1] W. Kang and J. McAuley, "Self-Attentive Sequential Recommendation," IEEE Int. Conf. on Data Mining, pp. 197–206, November 2018.
[2] Y. Zhang, B. Yang, H. Liu and D. Li, "A Time-aware Self-attention based Neural Network Model for Sequential Recommendation," Appl. Soft. Comput, vol. 133, pp. 109894, January 2023.
[3] J. Duan, P. Zhang, R. Qiu and Z. Huang, "Long Short-term Enhanced Memory for Sequential Recommendation," World Wide Web, vol. 26, pp. 561–583, May 2022.
[4] Y. Zheng, C. Gao, J. Chang, Y. Niu, Y. Song, D. Jin, et al, "Disentangling Long and Short-Term Interests for Recommendation," Proc. ACM Web Conf, pp. 2256–2267, April 2022.
[5] Y. Zhang, B. Yang, R. Mao and Q. Li, "MGT: Multi-granularity Transformer Leveraging Multi-level Relation for Sequential Recommendation," Expert Syst. Appl, Vol. 238, pp. 121808, March 2024.

[6] J. Wu, B. Yang, R. Mao and Q. Li, "Popularity-aware Sequential Recommendation with User Desire," Expert Syst. Appl, Vol. 237, pp. 121429, March 2024.

[7] H. Xu, B. Yang, X. Liu, "Reverse-graph Enhanced Graph Neural Networks for Session-based Recommendation, " Expert Syst. Appl, Vol. 245, pp. 122995, 1 July 2024.

[8] S. Rendle, C. Freudenthaler and L. Schmidt-Thieme, "Factorizing Personalized Markov Chains for Next-basket Recommendation," World Wide Web Conf, pp. 811–820, April 2010.

[9] R. He and J. McAuley, "Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation," IEEE Int. Conf. on Data Mining, pp. 191–200, December 2016.

[10] B. Hidasi, A. Karatzoglou, L. Baltrunas and D. Tikk,"Session-based Recommendations with Recurrent Neural Networks," ACM Int. Conf. Proc. Ser, pp. 1–10, May 2016.

[11] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian and J. Ma, "Neural Attentive Session-based Recommendation," Int. Conf. Inf. Knowledge Manage, pp. 1419–1428, November 2017.

[12] J. Tang and K. Wang, "Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding," Proc. ACM Int. Conf. on Web Search and Data Mining, pp. 565–573, February 2018.

[13] F. Yuan, A. Karatzoglou, L. Arapakis, J. Jose and X. He, "A Simple Convolutional Generative Network for Next Item Recommendation," Proc. ACM Int. Conf. on Web Search and Data Mining, pp. 582–590, January 2019.

[14] X. Xie, F. Sun, Z. Liu, S. Wu, J. Gao, J. Zhang, et al. "Contrastive Learning for Sequential Recommendation," IEEE Int. Conf. on Data Engi, pp. 1259—1273, August 2022.

[15] Y. Chen, Z. Liu, J. Li, J. McAuley, C. Xiong, "Intent Contrastive Learning for Sequential Recommendation," Proc. ACM Web Conf, pp. 2172–2182, April 2022.

[16] Z. Wei, N. Wu, F. Li, K. Wang and W. Zhang, "A Momentum Contrastive Learning Framework for Sequential Recommendation," Expert Syst. Appl, vol. 223, pp. 119911, August 2023.

[17] Z. Zhang, B. Yang, H. Xu, and W. Hu, "Multi-level Category-aware Graph Neural Network for Session-based recommendation," Expert Syst. Appl, Vol. 242, pp. 122773, May 2024.

[18] H. Liu, Z. Deng, L. Wang, J. Peng and S. Feng, "Distribution-based Learnable Filters with Side Information for Sequential Recommendation," Proc. ACM Conf. on Recom. Syst, pp. 78–88, September 2023.

[19] H. Xu, B. Yang, X. Liu, W. Fan and Q. Li, "Category-aware Multirelation Heterogeneous Graph Neural Networks for session-based recommendation," Knowl. Based Syst, vol. 251, pp. 109246, September 2022.

[20] S. Rendle, C. Freudenthaler, Z. Gantner and L. Schmidt-Thieme, "BPR: Bayesian Personalized Ranking from Implicit Feedback," ArXiv Preprint arXiv:1205.2618, May 2012.

[21] F. Sun, J, Liu, J. Wu, C. Pei, X. Lin, W. Ou, et al, "BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer," Int Conf. Inf. Knowledge Manage, pp. 1441–1450, November 2019.

[22] D. Kingma and J Ba, "Adam: A Method for Stochastic Optimization," ArXiv Preprint arXiv: 1412.6980, December 2014.

[23] T. Chen, S. Kornblith, M. Norouzi and G. Hinton. "A Simple Framework for Contrastive Learning of Visual Representations," Proc. Int. Conf. Machine Learning, pp. 1597–1607, July 2020.

[24] K. He, H. Fan, Y. Wu, S. Xie and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning," Proc. IEEE Conf. Comput. Vision and Pattern and Recog, pp. 9729–9738, June 2020.

[25] K. Zhou, B. Zhang, W. Zhao and J. Wen, "Debiased Contrastive Learning of Unsupervised Sentence Representations," Proc. Annual Meeting of the Asso. for Comput. Lingu, pp. 6120–6130, May 2022.

[26] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, et al, "Self-supervised Graph Learning for Recommendation," Int. ACM SIGIR Conf. Res. Dev. Inf. Retr, pp. 726–735, July 2021.

[27] T. Yao, X. Yi, D. Cheng, F. Yu, T. Chen, A. Menon, et al, "Self-supervised Learning for Large-scale Item Recommendations," Int Conf. Inf. Knowledge Manage, pp. 4321–4330, November 2021.

[28] Y. Zhang, G. Yin, Y. Dong and L. Zhang, "Contrastive Learning with Frequency Domain for Sequential Recommendation," Appl. Soft. Comput, vol. 144, pp. 110481, September 2023.