

Full Length Article

Multi-level sequence denoising with cross-signal contrastive learning for sequential recommendation

Xiaofei Zhu^{a,*}, Liang Li^a, Weidong Liu^b, Xin Luo^c^a College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China^b China Mobile Research Institute, China Mobile, Beijing 100053, China^c College of Computer and Information Science, Southwest University, Chongqing 400715, China

ARTICLE INFO

Keywords:

Recommender systems
 Sequential recommendation
 Sequence denoising
 Curriculum learning

ABSTRACT

Sequential recommender systems (SRSs) aim to suggest next item for a user based on her historical interaction sequences. Recently, many research efforts have been devoted to attenuate the influence of noisy items in sequences by either assigning them with lower attention weights or discarding them directly. The major limitation of these methods is that the former would still prone to overfit noisy items while the latter may overlook informative items. To the end, in this paper, we propose a novel model named Multi-level Sequences Denoising with Cross-signal Contrastive Learning (MSDCCL) for sequential recommendation. To be specific, we first introduce a target-aware user interest extractor to simultaneously capture users' long and short term interest with the guidance of target items. Then, we develop a multi-level sequence denoising module to alleviate the impact of noisy items by employing both soft and hard signal denoising strategies. Additionally, we extend existing curriculum learning by simulating the learning pattern of human beings. It is worth noting that our proposed model can be seamlessly integrated with a majority of existing recommendation models and significantly boost their effectiveness. Experimental studies on five public datasets are conducted and the results demonstrate that the proposed MSDCCL is superior to the state-of-the-art baselines. The source code is publicly available at <https://github.com/lalunex/MSDCCL/tree/main>.

1. Introduction

Sequential Recommender Systems (SRS) have emerged as a challenging research field and play a critical role in present e-commerce and social media platforms (Chen, Liu, Li, McAuley, & Xiong, 2023; Wang, Liu, Wang, Wang, & Li, 2023; Xie et al., 2022), with the aim of recommending the next item based on users' historical sequences. The core problem of SRS is how to effectively capture the sequential patterns of historical user behaviors. To this end, many kinds of methods based on deep neural networks have been proposed. For example, some works propose to leverage recurrent neural networks (RNNs) (Hidasi, Karatzoglou, Baltrunas, & Tikk, 2016; Orvieto et al., 2023), convolutional neural networks (CNNs) (Sudarsan & Polash, 2023; Tang & Wang, 2018), Transformer (Ma et al., 2023), and Graph Neural Networks (GNNs) (Hao et al., 2023) to model the sequential patterns. Despite these methods have achieved promising performance, they have a major limitation that user sequences may contain some noisy items (e.g., misclick (Tolomei, Lalmas, Farahat, & Haines, 2019) and malicious false interactions (Zhang, Li, Ding, & Gao, 2020), which

undoubtedly pose significant challenges for extracting user preference and making recommendations.

In recent years, many research efforts have been devoted to handle the noisy issue in users' historical sequences. One research line is to mitigate the impact of noise within sequences in a "soft" manner by utilizing the attention mechanism (Li et al., 2021; Li, Wang, & McAuley, 2020) or filtering algorithms (Zhou, Yu, Zhao, & Wen, 2022). These methods attempt to reduce the influence of noisy items by assigning them lower attention weights when learning representations of users' historical sequences. DSAN (Yuan, Song, Sun, Wang, & Zhao, 2021) introduces a virtual target item and takes it as the query vector for attention weight assignment. AC-TSR (Zhou et al., 2023) introduces two calibrators, i.e., a spatial calibrator and an adversarial calibrator, to assign the attention weights via utilizing spatial relationships and each item's contribution for prediction. FMLP-Rec (Zhou et al., 2022) applies fast fourier transform to convert items into the frequency domain and utilize a low-pass filter to alleviate noisy items in the frequency domain. Although these research efforts have achieved encouraging

* Corresponding author.

E-mail addresses: zxf@cqut.edu.cn (X. Zhu), liliang@stu.cqut.edu.cn (L. Li), liuweidong@chinamobile.com (W. Liu), luoxin@swu.edu.cn (X. Luo).<https://doi.org/10.1016/j.neunet.2024.106480>

Received 28 January 2024; Received in revised form 13 June 2024; Accepted 18 June 2024

Available online 29 June 2024

0893-6080/© 2024 Elsevier Ltd. All rights reserved, including those for text and data mining, AI training, and similar technologies.

results, they still suffer from noisy items in sequences (Zhang et al., 2022). Another research line proposes to explicitly remove noisy item in a “hard” manner (Sun, Wang, Sun, & Yang, 2021; Zhang et al., 2022). BERD (Sun et al., 2021) proposes to handle unreliable items in sequences by modeling their corresponding losses and uncertainties via a Gaussian distribution. HSD (Zhang et al., 2022) combines two levels of signals, i.e., the user-level signal and the sequence-level signal, to identify inherent noisy items in sequences. One major limitation of these methods is that they would ignore relevant items in sequences during the denoising process, which will lead to inferior performance.

To address the issues mentioned above, we propose a novel method called Multi-level Sequence Denoising with Cross-signal Contrastive Learning (MSDCCL) for sequential recommendation. The main idea of our model is to handle noisy items in user interaction sequences by simultaneously leveraging both “soft” and “hard” denoising strategies, where each of them mutually guide the learning of the other. Specifically, we develop a soft- and hard-level sequence denoising module which consists of two sub-modules, i.e., a soft-level denoising sub-module and a hard-level denoising sub-module. The former alleviates noisy items by assigning lower attention weights, and the latter eliminates irrelevant items identified by using a Gumbel-Softmax function (Chaudhary & Singh, 2023; Wang, Jia, Guo & Liu, 2023). It is worth noting that we also introduce a cross-signal contrastive learning layer to allow guidance informative exchange between the two sub-modules. To the best of our knowledge, this is the first model that can comprehensively explore the benefits of both “soft” and “hard” denoising strategies for sequential recommendation. In addition, we incorporate a target-aware user interest extractor to model long- and short-term user interests, where a transformer sub-module is utilized to capture the long-term user interest and a target-aware convolutional sequence embedding sub-module is developed to learn effective short-term user interest. At last, we extend existing curriculum learning (Wang, Pan, et al., 2023; Xu, Yuan, Mo, Song, & Li, 2024) by utilizing a S-shape function to simulate the learning process of human beings. To be specific, we separate “difficult” training samples into low- and high-speed learning zones, and train our model with fewer samples in the low-speed learning zone and more samples in the high-speed learning zone.

Extensive experiments have been conducted to examine the performance of our proposed model MSDCCL on five public benchmark datasets, i.e., ML-100k, Beauty, Sports, Yelp and ML-1M. The results demonstrate that MSDCCL significantly outperforms the state-of-the-art methods. For example, the relative performance improvements of MSDCCL over the two best performing baselines (i.e., HSD+BERT4Rec and AC-BERT4Rec) are 100.00% and 233.96% in terms of HR@5 on the dataset ML-100k, respectively. We further conduct an ablation study to verify the contribution of each component in our model, and the results show that removing each of them will lead to a considerable performance degradation. At last, we investigate the influence of the S-shape curriculum learning, and the results suggest that MSDCCL equipped with S-shape increment performs better than its counterpart equipped with linear increment. We summarize the main contributions of this paper as follows:

- We propose a novel denoising model for the task of sequential recommendation, which simultaneously attenuates the denoising issue in the sequence via both soft and hard denoising strategies.
- We design a target-aware user interest extractor to effectively capture both user long-term and short-term interest.
- We extend existing curriculum learning by simulating the learning pattern of human beings.
- Extensive experiments have been conducted on five widely used datasets, including ML-100k, ML-1M, Beauty, Sports and Yelp. The results show that the proposed MSDCCL is superior to all state-of-the-art baseline methods.

The rest of the paper is organized as follows. Section 2 gives the related work. Section 3 provides the preliminaries. We introduce the details of our proposed approach MSDCCL in Section 4, and present the experimental results and analysis in Section 5. Section 6 draws the conclusions of this paper.

2. Related work

2.1. Sequential recommendation

Sequential recommendation aims at extracting a user’s interest from her historical interactions and then predicting the next item that she is most interested in. Some pioneer works (Rendle, Freudenthaler, & Schmidt-Thieme, 2010) utilize Markov Chains (MCs) to capture dynamic changes in user interest. With the tremendous success of deep learning, in recent years, researchers resort to developing various deep neural networks based methods for sequential recommendation and have achieved encouraging progress. Hidasi et al. (2016) first propose to employ recurrent neural networks (RNNs) for sequential recommendation and introduce a novel ranking loss function for model training. Liu et al. (2016) model user sequential behaviors by leveraging the contextual information. They extend the conventional RNN models by introducing variable input matrix and transition matrix. Tang et al. (2018) utilize convolutional filters to extract sequential patterns and develop a convolutional sequence embedding recommendation model which assigns larger impact to more recent items. Yuan et al. (2019) extract both short and long-range item dependency to learn high-level representation, and increase the receptive fields by employing a stack of holed convolutional layers. Wu et al. (2019) model session sequences as graph-structured data and apply graph neural networks (GNNs) to capture complex transition patterns between items. Wang et al. (2020) further consider both the transition information and the corresponding temporal dynamics of items to learn accurate representations. They utilize external knowledge graph to capture the relation between items, and introduce a dynamic time kernel function to merge these representations.

2.2. Contrastive learning

Contrastive Learning (CL) (Huang, Wang, He, & Yin, 2022) has been successfully used in the field of recommendation, which attempts to extract useful self-supervised signals from user behavior sequences. Many efforts have been conducted to apply contrastive learning to improve the recommendation performance via augmenting the user-item bipartite graph. For example, Wu et al. (2021) utilize dropout-based structure perturbation to augment the user-item bipartite graph, and maximize the representation consistency between same nodes in different augmented graphs. Different to the dropout-based augmentation, Yu et al. (2022) propose a noise-based augmentation by constructing contrastive views via incorporating different random noises to the original representations. They aim to regularize the embedding space towards a uniform distribution.

Some other works attempt to apply contrastive learning to yield better performance for sequential recommendation. For example, Xia et al. (2021) introduce the hypergraph representation learning to alleviate the sparse issue of the session data, and provide strong self-supervision signals for recommendation. Qiu et al. (2021) develop a multi-instance noise contrastive estimation, which extends the noise contrastive estimation from a single positive sample to a multi-instance variant. Xie et al. (2022) employ stochastic data augmentation to transform each user interaction sequence into two correlated views of the sequence, and explore contrastive learning on user interaction sequences to boost sequential recommendation performance. Qiu et al. (2022) incorporate the target item to provide supervision signals for positive sampling, and design a contrastive regularization to improve the representation distribution. Chen et al. (2022) leverage unlabeled user behavior sequences

to capture users' intent distribution, and propose an intent contrastive learning module to maintain the agreement between a sequence and its corresponding intent. To reduce the false positive and false negative samples, Wang et al. (2022) employ explanation methods to derive positive and negative sequences based the importance of items in a user sequence. Duan et al. (2023) leverage the same user's short-term and long-term interaction sequences as positive samples, and other randomly selected interaction sequences as negative samples. Zhu et al. (2024) first learn a user's low-level and high-level preference interest, and then take the corresponding preference interest between them as positive samples, and others as negative samples.

The key difference between our method and existing models is that we propose a novel cross-signal (i.e., the generated soft and hard supervision signals) to generate the positive and negative samples. Specifically, we first utilize the generated hard supervision signals to identify noisy and relevant items. Then, we take these pairs between the user interest representation (i.e., generated via the soft supervision signals) and the noisy items as negative samples, and pairs between it and the relevant items as positive samples.

2.3. Denoising methods

Due to the existence of noise within users' historical interactions (Quan et al., 2023; Wang, Gao, et al., 2023; Zhang, Chen, Zhao, Han, & Li, 2023), e.g., accidental interactions, a surge of works have been proposed to deal with noisy items in sequences. These methods can be roughly grouped into two categories, i.e., soft-denoising category and hard-denoising category. The methods in the soft-denoising category mainly depend the self-attention mechanism, which attempts to assign lower attention weights to noisy items in order to alleviate their influence. Li et al. (2020) consider the influence of both the position of each item as well as the time interval between items on prediction and propose a time-aware self-attention mechanism for recommendation. Luo et al. (2020) first identify neighborhood sessions of the current session and then utilize the self-attention network to assign weights to collaborative items. Yuan et al. (2021) design a self-attention network to obtain target embedding and apply a vanilla attention network to estimate the importance of items in sequences. An adaptively sparse transformation function is then incorporated to eliminate the influence of noisy items in sequences. Zhou et al. (2023) improve the estimation of attention weights for items in a sequence by developing a spatial calibrator and an adversarial calibrator. The former is designed to adjust attention weights based on spatial information among items, and the latter is used to modify attention weights by exploring the contribution of each item. However, these methods would still assign attention weights to less relevant or irrelevant items in sequences, leading to sub-optimal recommendation performance.

The methods in the hard-denoising category aim to explicitly drop noisy items in sequences. Tong et al. (2021) employ reinforcement learning to determine the relevance of each item in sequences. They investigate sequential patterns for the policy learning process and formulate the denoising problem as a Markov Decision Process. Chen et al. (2022) propose a denoising strategy by getting rid of noisy attentions. They introduce a trainable binary mask in each self-attention layer which assigns zero attention scores to noisy items. This method remains the architectures of transformers, while changing the attention distributions. Zhang et al. (2022) propose a hierarchical sequence denoising method by introducing two types of signals, i.e., user-level intent signals and sequence-level context signals, to identify inconsistency items in sequences. Despite the promising performance achieved by these methods, identifying irrelevant items in sequences is still challenging and relevant information would be discarded during the denoising process. Different with previous works, we propose to combine both soft and hard denoising strategies in a unified model, where the two denoising strategies will guide the denoising process of each other.

3. Preliminaries

In our denoising model, we have a collection of users and items, denoted as \mathcal{U} and \mathcal{V} , correspondingly, where $u \in \mathcal{U}$ stands for a user and $v \in \mathcal{V}$ stands for an item. The quantity of users and items in the collection is indicated as $|\mathcal{U}|$ and $|\mathcal{V}|$, respectively. Our model takes the collection of users' historical interactions $S = \{S^{u_1}, S^{u_2}, \dots, S^{u_{|\mathcal{U}|}}\}$ as input. For a particular user, her interaction sequence is represented as $S = [s_1, s_2, \dots, s_n]$, where n indicates its length, and s_i denotes the i -th item that has been interacted. The aim of sequential recommendation is to predict the next item that the user is most likely to engage with at the $(n+1)$ th step, depicted as $p(s_{n+1}|s_{1:n})$.

4. Methodology

The architecture of our proposed model MSDCCL is illustrated in Fig. 1. It mainly consists of four components: embedding layer, target-aware user interest extractor, soft- and hard-level sequence denoising, and prediction layer.

4.1. Embedding layer

In the embedding layer, $\mathbf{M}_I \in \mathbb{R}^{|\mathcal{V}| \times d}$ denotes a dense, lower-dimensional item embedding matrix, where d represents the size of embedding vectors. The corresponding item embedding matrix $\mathbf{H} \in \mathbb{R}^{n \times d}$ for a user's history interaction sequence S can be obtained by performing a look-up operation on \mathbf{M}_I . To capture the temporal influence of each item in S , we also incorporate a learnable position embedding matrix $\mathbf{P} \in \mathbb{R}^{n \times d}$. Therefore, for an item s_i in S , we have $h_i \in \mathbf{H}$ and $p_i \in \mathbf{P}$ to represent its item embedding and corresponding position embedding, respectively.

4.2. Target-aware user interest extractor

User historical interactions reflect the evolution of her interests, leading to different interests between long- and short-term interactions (Lv, Zhuang, Luo, Li, & Zha, 2020; Zheng, Liu, Li, & Wu, 2021). To represent the long-term user interest, we utilize a transformer encoder to model the whole historical interactions. Previous research has emphasized the importance of modeling the short-term user interest based on her recent interactions. In this regard, it is common to utilize the last m items in the historical interactions to capture the short-term user interest (Liu, Zeng, Mokhosi, & Zhang, 2018; Zheng et al., 2022). In contrast, some researches employ the next item to reflect the short-term user interest (Lin et al., 2022; Zhang et al., 2022). Due to potential shifts of the short-term user interest, we integrate those two approaches during the training phase to improve its stability. As the target item remains unobservable during the validation and testing phases, we exclusively use the last m items in the sequence to model the short-term user interest. After acquiring the representations of user's long- and short-term interest, we utilize a feedforward neural network to fuse them into a comprehensive representation of user interest.

4.2.1. Long-term user interests

Inspired by the works of Xu, Peng, et al. (2024), Zhou et al. (2020), our model incorporates a transformer encoder to encode users' historical interactions. It consists of multiple identical layers that comprise of multi-head self-attention and feed-forward neural networks, which can be formulated as follows:

$$\mathbf{H}^l = \text{Transformer}(\mathbf{H}^{l-1} + \mathbf{P}), \quad \forall l \in [1, \dots, L], \quad (1)$$

where \mathbf{H} and \mathbf{P} represent the item embedding matrix and the corresponding position embedding matrix, respectively. To obtain the final output of the transformer encoder, we retrieve the hidden representation matrix $\mathbf{H}^L = [h_1^L, h_2^L, \dots, h_n^L] \in \mathbb{R}^{n \times d}$ from the last layer L . For simplicity, we utilize the hidden representation of last item h_n^L as the long-term user interest $e_l^u \in \mathbb{R}^d$.

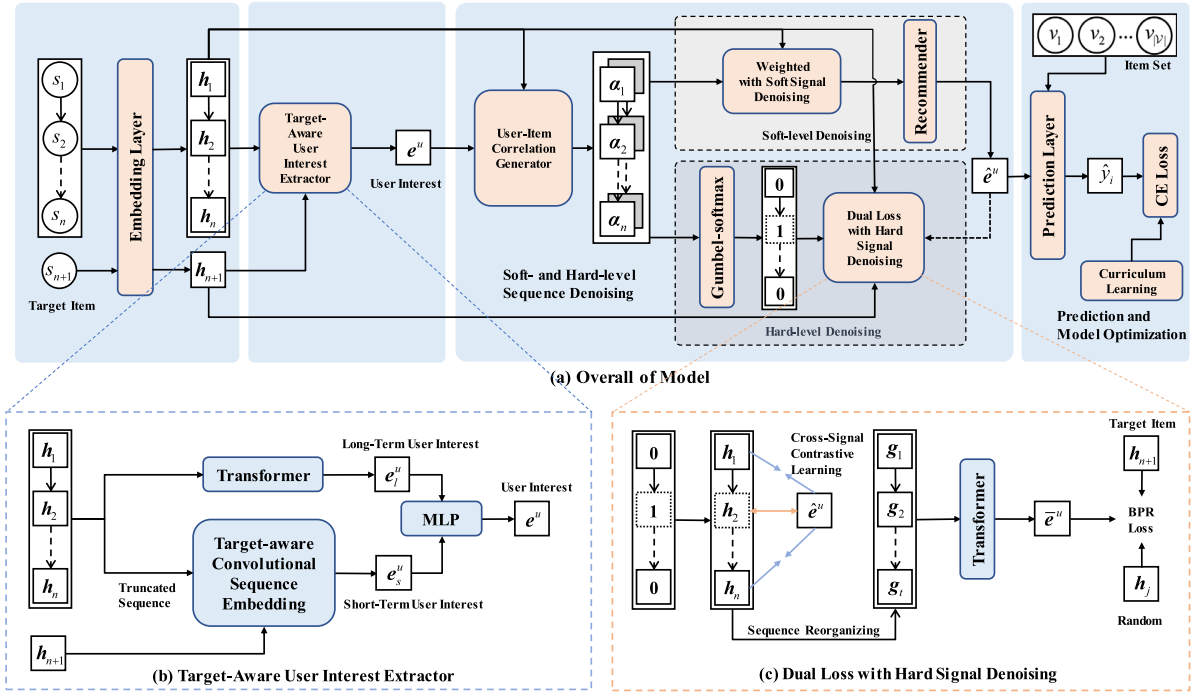


Fig. 1. The architecture of our proposed model MSDCCL.

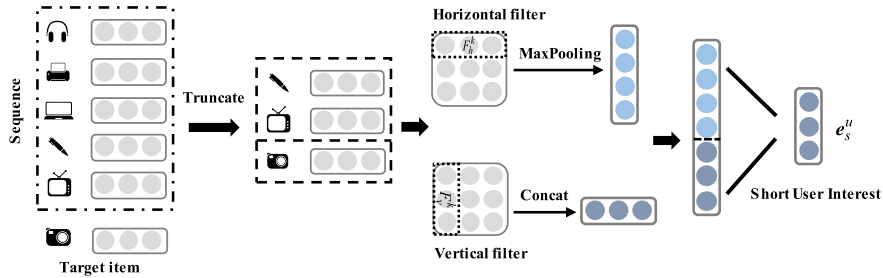


Fig. 2. Target-aware convolutional sequence embedding.

4.2.2. Short-term user interests

Based on previous discussions, we propose to integrate both the last m items and the next item to learn a high quality short-term user interest representation. This integration entails jointly modeling the actual target item and the last m items, which is formulated as follows:

$$H' = \text{Truncate}(H, m), \quad (2)$$

$$c_h^k = \text{Conv}_h^k([H' \parallel h_{n+1}], F_h^k), \forall k \in [1, \dots, z(m+1)], \quad (3)$$

$$c_v^k = \text{Conv}_v^k([H' \parallel h_{n+1}], F_v^k), \forall k \in [1, \dots, z'], \quad (4)$$

where $\text{Truncate}(\cdot)$ is a function which truncates the last m items from the sequence S , h_{n+1} is the target item representation, and \parallel denotes the concatenation operation. It is important to note that c_h^k and c_v^k are obtained by applying convolution with a stride of 1, where F_h^k and F_v^k stand the horizontal and vertical convolution kernel, respectively. Meanwhile, z and z' represent the number of convolution kernels of the same shape. The specific convolution process is shown in Fig. 2. Subsequently, we fuse $c_h \in \mathbb{R}^{z(m+1)}$ and $c_v \in \mathbb{R}^{z'd}$ to obtain the short-term user interest as follows:

$$c_h = \text{MaxPooling}(\text{ReLU}(c_h^1, c_h^2, \dots, c_h^{z(m+1)})), \quad (5)$$

$$c_v = [c_v^1 \parallel c_v^2 \parallel \dots \parallel c_v^{z'}], \quad (6)$$

$$e_s^u = \text{MLP}([c_h \parallel c_v]). \quad (7)$$

4.2.3. Interests fusion

After obtaining the long- and short-term interests of user, we employ a feedforward network to combine the two interest representations, yielding the final user interest $e^u \in \mathbb{R}^d$, which is formulated as follows:

$$e^u = (\text{ReLU}(e_s^u W_1 + e_l^u W_1)) W_2 + b_1 \quad (8)$$

where $W_1, W_2 \in \mathbb{R}^{d \times d}$ and $b_1 \in \mathbb{R}^d$ denote the trainable parameters.

4.3. Soft- and hard-level sequence denoising

4.3.1. User-item correlation generator

Users' historical interaction sequences inevitably contain noise (Tolomei et al., 2019) and in some cases, even malicious false interactions (Zhang et al., 2020). The noise within interaction sequences can potentially mislead the model, resulting in sub-optimal user representations. To identify noisy items in a sequence S , one solution is to employ the attention mechanism as a discriminator and use e^u as a query to assign different weights $\alpha_i \in \mathbb{R}^2$ to distinct items in S . It is worth noting that the first and second dimensions of α_i respectively denote the relevance and irrelevance of item s_i to user interest e^u , which can be formulated as follows:

$$h'_i = h_i W_3, \quad (9)$$

$$\alpha_i = \sigma([h'_i \parallel e^u \parallel h'_i - e^u \parallel h'_i \odot e^u] W_4), \quad (10)$$

where $\mathbf{W}_3 \in \mathbb{R}^{d \times d}$ and $\mathbf{W}_4 \in \mathbb{R}^{4d \times 2}$ are the trainable parametric matrices, \odot and $\sigma(\cdot)$ represent the element-wise product and the sigmoid activation function, respectively.

4.3.2. Soft-level denoising

Similar to previous studies (Li et al., 2020; Xu et al., 2019), we introduce a soft-level denoising module by carrying out a soft signal denoising layer on the sequence S based on the attention mechanism, which is followed by a recommender layer.

Weighted with Soft Signal Denoising. We employ the first dimension of $\alpha_i \in \mathbb{R}^2$ to represent the relevance between the i th item and the user interest, and utilize the normalized weights $\{\alpha_i^0\}_{i=1}^n$ to assign weights to items within the sequence. After that, we obtain the new item representation $\hat{\mathbf{h}}_i \in \mathbb{R}^d$:

$$\alpha_i^0 = \frac{\exp(\alpha_i^0)}{\sum_{j=1}^n \exp(\alpha_j^0)}, \quad (11)$$

$$\hat{\mathbf{h}}_i = \alpha_i^0 \mathbf{h}_i \mathbf{W}_5, \quad (12)$$

where $\mathbf{W}_5 \in \mathbb{R}^{d \times d}$ is a trainable parametric matrix, and $\hat{\mathbf{H}} = [\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2, \dots, \hat{\mathbf{h}}_n] \in \mathbb{R}^{n \times d}$ is the soft denoised sequence representation.

Recommender. The recommender serves as a sequence encoder, which can be any mainstream sequence recommendation models. It takes $\hat{\mathbf{H}}$ as input and generates the user representation $\hat{\mathbf{e}}^u$, which can be formulated as follows:

$$\hat{\mathbf{e}}^u = \mathbf{F}(\hat{\mathbf{H}}), \quad (13)$$

where $\mathbf{F}(\cdot)$ denotes a recommender utilized for sequence representation learning, e.g., BERT4Rec (Sun et al., 2019), SASRec (Kang & McAuley, 2018), etc.

4.3.3. Hard-level denoising

We can predict the next item by solely relying on $\hat{\mathbf{e}}^u$. However, the influence of noisy items cannot be entirely eliminated based on the soft-level denoising since it still assigns some weights to them and leading to sub-optimal recommendation performance. To deal with this issue, we propose to further incorporate a hard-level denoising module to improve the learning of user representation. To achieve this, we introduce contrastive learning to enhance the robustness of the user interest representation. Nevertheless, the generated hard signals, serving as pseudo-labels, are not fully accurate in detecting noise items due to there is no supervised signals. To address this challenge, we employ the Bayesian Personalized Ranking (BPR) loss (Qin, Ju, Wu, Luo, & Zhang, 2024; Rendle, Freudenthaler, Gantner, & Schmidt-Thieme, 2009) to preserve the quality of the generated hard signals.

Gumbel-softmax. Since $\alpha_i \in \mathbb{R}^2$ serves as a discriminative metric to determine whether an item is noise or not, it can be applied to guide the generation of the hard signals. Due to the need to obtain a binary value and allow gradient backpropagation, we opt for the Gumbel-softmax function to satisfy the requirement, which can be formulated as follows:

$$\tilde{\alpha}_i = \text{Gumbel-softmax}(\alpha_i, \tau), \quad (14)$$

$$\tilde{\alpha}_i^1 = \frac{\exp(\log(\alpha_i^1) + g_1) / \tau}{\sum_{j=0}^1 \exp(\log(\alpha_i^j) + g_j) / \tau}, \quad (15)$$

where g_j represents a perturbation sampled from the Gumbel distribution, used to enhance the model's robustness. Additionally, $\tau > 0$ serves as a temperature parameter that regulates the selection distribution. When $\tau \rightarrow 0$, $\tilde{\alpha}_i$ approximates a one-hot vector. When $\tau \rightarrow \infty$, $\tilde{\alpha}_i$ approximates a uniform distribution. When $\tau \rightarrow 1$, the Gumbel-softmax function gradually converges to the standard softmax function. Note that we leverage the second dimension of $\tilde{\alpha}_i = [\tilde{\alpha}_i^0, \tilde{\alpha}_i^1] \in [0, 1]$ as the hard signal because it denotes the irrelevance between the item representation and the user interest representation. Specifically, if $\tilde{\alpha}_i^1 = 1$, it means that s_i is a noisy item, and otherwise s_i is a relevant item.

Dual Loss with Hard Signal Denoising. In this component, we focus on the task of implementing hard denoising for sequences. It initially uses the generated hard signals $\{\tilde{\alpha}_i^1\}_{i=1}^n$ to identify noisy items within the sequence. Subsequently, we propose a novel cross-signal contrastive learning to enhance the user interest representation $\hat{\mathbf{e}}^u$, which is generated via the soft denoising. In Fig. 1(c), \mathbf{h}_2 corresponds to a noisy item, while \mathbf{h}_1 corresponds to a relevant item. The contrastive learning objective is to maximize the correlation between positive sample pairs $(\hat{\mathbf{e}}^u, \mathbf{h}_1)$ and minimize the correlation between negative sample pairs $(\hat{\mathbf{e}}^u, \mathbf{h}_2)$:

$$\mathcal{L}_{SCL} = -\frac{1}{|S^+|} \sum_{s_i \in S^+} \log \frac{\exp(\text{sim}(\hat{\mathbf{e}}^u, \mathbf{h}_i) / \tau)}{\sum_{s_j \in S} \exp(\text{sim}(\hat{\mathbf{e}}^u, \mathbf{h}_j) / \tau)} \quad (16)$$

where $\text{sim}(\cdot)$ is a cosine similarity function, τ represents a temperature parameter, and $|S^+|$ represents the number of positive samples in the sequence S .

However, without supervised signals to ascertain whether items in the sequence are noise, the reliability of the generated hard signals $\{\tilde{\alpha}_i^1\}_{i=1}^n$ cannot be determined. To tackle this, we use the hard signals to explicit eliminate “noise” items from the original historical interactions, yielding a new sequence representation $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_t] \in \mathbb{R}^{t \times d}, t \leq n$. Next, we employ a transformer encoder to encode \mathbf{G} in order to acquire the hard denoised user representation $\bar{\mathbf{e}}^u$. The process can be formalized as follows:

$$\mathbf{G} = \text{Reorganize}((\mathbf{1} - \tilde{\alpha}^1) \cdot \mathbf{H}), \quad (17)$$

$$\bar{\mathbf{e}}^u = \text{Transformer}(\mathbf{G}), \quad (18)$$

where $\tilde{\alpha}^1 = [\tilde{\alpha}_1^1, \tilde{\alpha}_2^1, \dots, \tilde{\alpha}_n^1] \in \mathbb{R}^d$ denotes the generated hard denoising signal, and $\text{Reorganize}(\cdot)$ is a function to reorganize the sequence S after hard denoising.

After we obtain $\bar{\mathbf{e}}^u$, a Bayesian Personalized Ranking (BPR) loss is utilized to maintain the reliability of the generated hard denoised signals:

$$\mathcal{L}_{BPR} = -\frac{1}{|\mathcal{O}|} \sum_{(u,i,j) \in \mathcal{O}} \ln \sigma(\hat{r}_{ui} - \hat{r}_{uj}), \quad (19)$$

where \hat{r}_{ui} and \hat{r}_{uj} are the vector inner-product of the positive sample pair $(\bar{\mathbf{e}}^u, \mathbf{h}_{i+1}) \in \mathcal{R}^+$ and the negative sample pair $(\bar{\mathbf{e}}^u, \mathbf{h}_j) \in \mathcal{R}^-$, respectively. $\mathcal{O} = \{(u, i, j) \mid (u, i) \in \mathcal{R}^+, (u, j) \in \mathcal{R}^-\}$. Since a user's target item is unique, thus the positive sample for the user is \mathbf{h}_{n+1} , and the randomly sampled negative sample for the user is \mathbf{h}_j .

4.4. Prediction

After applying soft and hard denoising, we obtain a relatively noise-free representation of user interest $\bar{\mathbf{e}}^u$. Subsequently, within the entire item set space \mathcal{V} , we calculate the score between each item $v_i \in \mathcal{V}$ and the user's interest representation $\bar{\mathbf{e}}^u$, denoted as z_i , to recommend the next item. For each candidate item v_i , we can use the following formula to quantify its relevance to the sequence:

$$z_i = \bar{\mathbf{e}}^u \mathbf{h}_{v_i}^T, \quad (20)$$

then the predicted probability distribution of the next item within the entire item set space \mathcal{V} can be calculated as follows:

$$\hat{y}_i = \frac{\exp(z_i)}{\sum_{v_j \in \mathcal{V}} \exp(z_j)}. \quad (21)$$

Subsequently, we define the sequence recommendation task by minimizing the cross-entropy of the predicted results \hat{y}_i :

$$\mathcal{L}_{Rec} = -\sum_{i=1}^{|\mathcal{V}|} (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)), \quad (22)$$

where y_i denotes the i th one-hot encoding of the ground truth item.

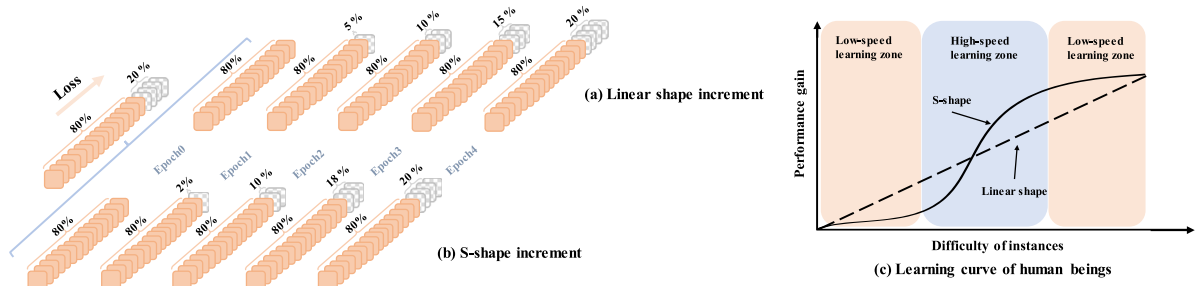


Fig. 3. S-shape curriculum learning vs. Linear shape curriculum learning.

Table 1
Statistics of the datasets.

| Dataset | #Sequence | #Users | #Items | #Avg.length | #Sparsity |
|---------|-----------|--------|--------|-------------|-----------|
| ML-100k | 99,287 | 944 | 1350 | 105.29 | 92.21% |
| Beauty | 198,502 | 22,364 | 12,102 | 8.88 | 99.93% |
| Sports | 296,337 | 35,599 | 18,358 | 8.32 | 99.95% |
| Yelp | 316,354 | 30,432 | 20,034 | 10.40 | 99.95% |
| ML-1M | 999,611 | 6041 | 3417 | 165.50 | 95.16% |

Finally, we use an end-to-end approach to minimize all above loss functions (i.e., Eqs. (16), (19), (22)) to achieve a high performance sequential recommendation model:

$$\text{Loss} = \frac{1}{|U|} \sum_{u=1}^{|U|} [\mathcal{L}_{\text{Rec}} + \lambda (\mathcal{L}_{\text{SCL}} + \mathcal{L}_{\text{BPR}})] + \beta \|\Theta\|_2^2, \quad (23)$$

where λ is a hyperparameter that balances the dual loss and the recommendation loss, β is a hyperparameter used to solve the overfitting problem of the model, and Θ is the total parameter set of the model.

4.5. S-shape curriculum learning

To boost the performance of our model, we further introduce the curriculum learning (Fu & Chen, 2023; Wang, Pan, et al., 2023; Xu, Yuan, et al., 2024). To be specific, we start to train our model from simple samples and gradually move to more difficult ones. According to Zhang et al. (2022) and Wang et al. (2021), we rank the loss values of all instances in the mini-batch in ascending order. Following the 20/80 principle (Sun et al., 2020), we divide the mini-batch into “easy” instances (i.e., those with lower loss) and “difficult” instances (i.e., those with higher loss), where “easy” instances and “difficult” instances accounting for 80% and 20%, respectively. During the curriculum learning process, the training instances of the model include all the “easy” instances together with the μ percent of “difficult” instances, where μ gradually increases from 0, progressing with the iterations until epoch T reaches the limit M . Both T and M are predefined hyperparameters.

Existing curriculum learning methods usually assume the training process of increasing “difficult” instances in curriculum learning follows a linear pattern. As depicted in Fig. 3(a), in these methods, the “difficult” instances are fed into model training with a linear shape increment. However, different from them, we argue that increasing “difficult” instances should follow a S-shape increment, which is inspired by the learning pattern of human beings (Murre, 2014). As shown in Fig. 3(c), the learning performance of human beings are lower in the low-speed learning zone, but higher in the high-speed learning zone. Inspired by this phenomenon, we employ fewer “difficult” instances in the low-speed learning zones and more “difficult” instances in the high-speed learning zone to accommodate different learning stages. To be specific, we utilize a S-shape function (i.e., sigmoid function) to simulate the augmentation process, as depicted in Fig. 3(b).

5. Experiments

5.1. Experimental settings

5.1.1. Datasets

We conduct extensive experiments on five public benchmark datasets to evaluate the performance of our model. Table 1 shows the statistics of the datasets.

- **MovieLens¹**: MovieLens is a widely used dataset in the task of sequential recommendation. It contains both user ratings and reviews for movies. In our experiments, we adopt the 100k and 1M versions (named ML-100k and ML-1M, respectively). Note that the average sequence length of this dataset is relatively larger than others.
- **Amazon-Beauty and Sports²**: The dataset contains user historical purchases for a variety of products. Two representative subcategories (i.e., Beauty and Sports) are leveraged in the experiments. Different from MovieLens, users’ historical interaction sequences in this dataset is relative short (e.g., 8.88 and 8.32 items on average for Beauty and Sports, respectively).
- **Yelp³**: Yelp is another widely used dataset for sequential recommendation, which records user reviews for various restaurants and bars. As the original dataset is large, we only extract transaction records after 1 January 2019.

For all datasets, we keep users’ historical interaction sequences in a chronological manner. Following (Zhang et al., 2022; Zhou et al., 2022), we filter out inactive items with less than 5 interactions and inactive users with less than 5 items. The maximum sequence length for ML-1M is set to 200, and the others are set to 50. Finally, we employ the leave-one-out strategy to split data into training, validation and testing dataset.

5.1.2. Evaluation metrics

To evaluate the performance of all models, we employ a variety of widely used evaluation metrics (Zhang et al., 2022; Zhou et al., 2022), including *Hit Ratio* (HR@K), *Normalized Discounted Cumulative Gain* (NDCG@K), and *Mean Reciprocal Rank* (MRR@K). To be specific, we utilize HR@{5, 10, 20}, NDCG@{5, 10, 20} and MRR@{20} over the entire item set space to avoid the bias introduced by the sampling process (Cai, Wu, San, Wang, & Wang, 2021; Krichene & Rendle, 2020).

5.1.3. Competing models

To demonstrate the effectiveness of MSDCCL, we fuse it with six base models of sequential recommendation, and compare them with their counterparts. These base models are given as follows:

¹ <https://movielens.org/>

² <http://jmcauley.ucsd.edu/data/amazon>

³ <https://www.yelp.com/dataset>

Table 2

Performance comparison of different sequential recommendation methods with (w) or without (w/o) MSDCCL on the five datasets. The best score is in bold. All improvements are statistically significant (i.e., two-sided t-tests with $p < 0.05$).

| Dataset | Metric | GRU4Rec | | NARM | | STAMP | | Caser | | SASRec | | CL4SRec | | BERT4Rec | |
|---------|---------|---------|---------------|--------|---------------|--------|---------------|--------|---------------|---------------|---------------|---------|---------------|----------|---------------|
| | | w/o | w | w/o | w | w/o | w | w/o | w | w/o | w | w/o | w | w/o | w |
| ML-100k | HR@5 | 0.0191 | 0.0723 | 0.0180 | 0.0756 | 0.0201 | 0.0546 | 0.0212 | 0.0594 | 0.0191 | 0.0682 | 0.0289 | 0.0437 | 0.0191 | 0.0708 |
| | HR@10 | 0.0286 | 0.1249 | 0.0403 | 0.1317 | 0.0392 | 0.0943 | 0.0339 | 0.1051 | 0.0371 | 0.1203 | 0.0579 | 0.0777 | 0.0414 | 0.1270 |
| | HR@20 | 0.0594 | 0.2169 | 0.0657 | 0.2202 | 0.0700 | 0.1702 | 0.0679 | 0.1916 | 0.0764 | 0.2123 | 0.1107 | 0.1404 | 0.0912 | 0.2126 |
| | NDCG@5 | 0.0104 | 0.0453 | 0.0132 | 0.0470 | 0.0115 | 0.0331 | 0.0113 | 0.0362 | 0.0114 | 0.0437 | 0.0174 | 0.0267 | 0.0117 | 0.0449 |
| | NDCG@10 | 0.0134 | 0.0621 | 0.0202 | 0.0650 | 0.0176 | 0.0459 | 0.0153 | 0.0509 | 0.0172 | 0.0603 | 0.0265 | 0.0378 | 0.0189 | 0.0631 |
| | NDCG@20 | 0.0212 | 0.0821 | 0.0267 | 0.0871 | 0.0253 | 0.0648 | 0.0238 | 0.0726 | 0.0270 | 0.0834 | 0.0398 | 0.0536 | 0.0315 | 0.0845 |
| | MRR@20 | 0.0109 | 0.0494 | 0.0162 | 0.0509 | 0.0132 | 0.0365 | 0.0119 | 0.0409 | 0.0139 | 0.0487 | 0.0209 | 0.0301 | 0.0157 | 0.0498 |
| Beauty | HR@5 | 0.0077 | 0.0295 | 0.0120 | 0.0365 | 0.0080 | 0.0421 | 0.0072 | 0.0196 | 0.0242 | 0.0436 | 0.0208 | 0.0249 | 0.0060 | 0.0522 |
| | HR@10 | 0.0135 | 0.0481 | 0.0209 | 0.0565 | 0.0135 | 0.0638 | 0.0133 | 0.0340 | 0.0386 | 0.0632 | 0.0379 | 0.0434 | 0.0127 | 0.0714 |
| | HR@20 | 0.0256 | 0.0753 | 0.0367 | 0.0855 | 0.0231 | 0.0932 | 0.0235 | 0.0550 | 0.0561 | 0.0881 | 0.0618 | 0.0694 | 0.0204 | 0.0955 |
| | NDCG@5 | 0.0045 | 0.0194 | 0.0071 | 0.0243 | 0.0046 | 0.0285 | 0.0044 | 0.0120 | 0.0129 | 0.0307 | 0.0116 | 0.0156 | 0.0037 | 0.0378 |
| | NDCG@10 | 0.0064 | 0.0254 | 0.0099 | 0.0307 | 0.0064 | 0.0355 | 0.0064 | 0.0166 | 0.0175 | 0.0370 | 0.0171 | 0.0215 | 0.0059 | 0.0439 |
| | NDCG@20 | 0.0094 | 0.0322 | 0.0139 | 0.0380 | 0.0088 | 0.0429 | 0.0090 | 0.0219 | 0.0219 | 0.0434 | 0.0231 | 0.0280 | 0.0078 | 0.0500 |
| | MRR@20 | 0.0051 | 0.0204 | 0.0077 | 0.0249 | 0.0049 | 0.0289 | 0.0051 | 0.0128 | 0.0122 | 0.0307 | 0.0125 | 0.0167 | 0.0044 | 0.0372 |
| Sports | HR@5 | 0.0064 | 0.0176 | 0.0099 | 0.0199 | 0.0071 | 0.0260 | 0.0069 | 0.0119 | 0.0113 | 0.0258 | 0.0150 | 0.0270 | 0.0055 | 0.0271 |
| | HR@10 | 0.0114 | 0.0293 | 0.0138 | 0.0327 | 0.0123 | 0.0413 | 0.0115 | 0.0199 | 0.0175 | 0.0377 | 0.0256 | 0.0422 | 0.0104 | 0.0387 |
| | HR@20 | 0.0183 | 0.0465 | 0.0223 | 0.0517 | 0.0182 | 0.0629 | 0.0178 | 0.0329 | 0.0268 | 0.0546 | 0.0418 | 0.0638 | 0.0167 | 0.0553 |
| | NDCG@5 | 0.0035 | 0.0115 | 0.0058 | 0.0127 | 0.0046 | 0.0169 | 0.0046 | 0.0078 | 0.0059 | 0.0183 | 0.0088 | 0.0177 | 0.0036 | 0.0192 |
| | NDCG@10 | 0.0051 | 0.0153 | 0.0073 | 0.0169 | 0.0062 | 0.0218 | 0.0061 | 0.0103 | 0.0079 | 0.0221 | 0.0122 | 0.0226 | 0.0051 | 0.0229 |
| | NDCG@20 | 0.0068 | 0.0196 | 0.0094 | 0.0216 | 0.0077 | 0.0272 | 0.0077 | 0.0136 | 0.0102 | 0.0264 | 0.0163 | 0.0280 | 0.0067 | 0.0271 |
| | MRR@20 | 0.0036 | 0.0122 | 0.0059 | 0.0134 | 0.0048 | 0.0174 | 0.0049 | 0.0083 | 0.0055 | 0.0185 | 0.0093 | 0.0182 | 0.0040 | 0.0192 |
| Yelp | HR@5 | 0.0057 | 0.0215 | 0.0113 | 0.0248 | 0.0060 | 0.0234 | 0.0045 | 0.0224 | 0.0293 | 0.0212 | 0.0276 | 0.0307 | 0.0087 | 0.0243 |
| | HR@10 | 0.0102 | 0.0375 | 0.0187 | 0.0416 | 0.0099 | 0.0390 | 0.0084 | 0.0336 | 0.0352 | 0.0357 | 0.0425 | 0.0471 | 0.0159 | 0.0418 |
| | HR@20 | 0.0184 | 0.0631 | 0.0315 | 0.0686 | 0.0161 | 0.0645 | 0.0146 | 0.0517 | 0.0439 | 0.0584 | 0.0657 | 0.0711 | 0.0273 | 0.0695 |
| | NDCG@5 | 0.0034 | 0.0133 | 0.0075 | 0.0157 | 0.0038 | 0.0152 | 0.0028 | 0.0158 | 0.0251 | 0.0136 | 0.0185 | 0.0209 | 0.0054 | 0.0151 |
| | NDCG@10 | 0.0048 | 0.0184 | 0.0099 | 0.0211 | 0.0051 | 0.0202 | 0.0040 | 0.0194 | 0.0270 | 0.0183 | 0.0233 | 0.0261 | 0.0077 | 0.0207 |
| | NDCG@20 | 0.0068 | 0.0248 | 0.0131 | 0.0278 | 0.0066 | 0.0266 | 0.0055 | 0.0239 | 0.0292 | 0.0240 | 0.0291 | 0.0321 | 0.0105 | 0.0277 |
| | MRR@20 | 0.0037 | 0.0144 | 0.0081 | 0.0167 | 0.0040 | 0.0162 | 0.0031 | 0.0163 | 0.0250 | 0.0146 | 0.0191 | 0.0214 | 0.0060 | 0.0163 |
| ML-1M | HR@5 | 0.0194 | 0.1689 | 0.0151 | 0.1676 | 0.0232 | 0.1464 | 0.0104 | 0.1394 | 0.0397 | 0.1455 | 0.0369 | 0.0754 | 0.0224 | 0.1351 |
| | HR@10 | 0.0373 | 0.2473 | 0.0349 | 0.2457 | 0.0440 | 0.2145 | 0.0215 | 0.2158 | 0.0666 | 0.2251 | 0.0684 | 0.1294 | 0.0495 | 0.2095 |
| | HR@20 | 0.0690 | 0.3445 | 0.0591 | 0.3430 | 0.0677 | 0.3010 | 0.0589 | 0.3173 | 0.1007 | 0.3260 | 0.1204 | 0.2064 | 0.0980 | 0.3062 |
| | NDCG@5 | 0.0135 | 0.1139 | 0.0080 | 0.1124 | 0.0150 | 0.0993 | 0.0063 | 0.0899 | 0.0207 | 0.0936 | 0.0216 | 0.0471 | 0.0132 | 0.0873 |
| | NDCG@10 | 0.0190 | 0.1392 | 0.0144 | 0.1376 | 0.0218 | 0.1212 | 0.0099 | 0.1145 | 0.0294 | 0.1192 | 0.0317 | 0.0644 | 0.0218 | 0.1112 |
| | NDCG@20 | 0.0270 | 0.1636 | 0.0205 | 0.1621 | 0.0278 | 0.1430 | 0.0194 | 0.1401 | 0.0379 | 0.1447 | 0.0447 | 0.0837 | 0.0339 | 0.1356 |
| | MRR@20 | 0.0159 | 0.1129 | 0.0100 | 0.1113 | 0.0168 | 0.0988 | 0.0091 | 0.0908 | 0.0203 | 0.0940 | 0.0243 | 0.0501 | 0.0169 | 0.0881 |

- **GRU4Rec** (Hidasi et al., 2016). This method applies the gated recurrent unit (GRU) to model sequential data for recommendation. It modified the original GRU by introducing a tailored ranking loss function and session-parallel mini-batches.
- **NARM** (Li et al., 2017). It utilizes a hybrid encoder to capture the sequential behavior and the main purpose of a user in the current sequence. In addition, a bi-linear matching module is introduced to obtain recommendation scores.
- **STAMP** (Liu et al., 2018). This method proposes to capture both users' general interests and current interests. It leverages the long-term memory of the whole session as the general interests, and the short-term memory of the last-clicks as the current interests.
- **CASER** (Tang & Wang, 2018). It employs convolutional operation to capture sequential patterns and models users' recent items from both time and latent dimensions.
- **SASRec** (Kang & McAuley, 2018). It proposes a self-attention based sequential recommendation method, which utilizes self-attention mechanism to simultaneously capture both long-term semantics as well as a few important recent items.
- **CL4SRec** (Xie et al., 2022). It is a contrastive learning based sequential recommender which attempts to learn better sequence representations by leveraging the contrastive learning framework to obtain self-supervision signals.
- **BERT4Rec** (Sun et al., 2019). This method applies the deep bidirectional self-attention to model user behavior sequence, and utilizes the Cloze task as the objective in sequential recommendation.

In addition, we also compare the performance of our fused model (i.e., MSDCCL+BEAT4Rec) with the following state-of-the-art denoising models, including:

- **DSAN** (Yuan et al., 2021). It learns target embedding by exploring item-level interaction and correlation within a session. In addition, an adaptively sparse attention is used to mitigate the influence of irrelevant items.
- **FMLP-Rec** (Zhou et al., 2022). To obtain more robust sequence representations, it utilizes the fast fourier transform (FFT) (Gonzalez-Toral, Reviriego, Maestro, & Gao, 2018) and its inverse transform process to attenuate the effect of noise within a sequence.
- **HSD+BERT4Rec** (Zhang et al., 2022). It removes the noisy items in the original sequence by generating sequence-level and user-level signals, and learns better sequence representations by using the reassembled sequence.
- **AC-BERT4Rec** (Zhou et al., 2023). This model introduces a novel spatial calibrator and an adversarial calibrator to capture spatial relationships between items and redistribute attention weights based on the contributions of different items to predict the next item. In this model, we take BEAT4Rec as the backbone.

5.1.4. Implementation details

Similar to previous models (Yuan et al., 2021; Zhang et al., 2022), we set the embedding and mini-batch sizes for all models to 100 and 256, respectively. The learning rate of Adam optimizer is set to 10^{-3} . We tune the regularization hyperparameter β in $\{0, 10^{-3}, 10^{-4}\}$ on the validation set. The initial temperature parameter τ in the cross-signal

Table 3

Performance comparison between MSDCCL with the best base model (i.e., BERT4Rec) and the state-of-the-art denoising methods on the five datasets. The best score and the second best score in each column are in bold and underlined, respectively. OOM: Out of memory on 24GB DGX.

| Dataset | Model | HR@5 | HR@10 | HR@20 | NDCG@5 | NDCG@10 | NDCG@20 | MRR@20 |
|---------|------------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| ML-100k | DSAN (AAAI'21) | 0.0201 | 0.0435 | 0.0700 | 0.0115 | 0.0188 | 0.0254 | 0.0133 |
| | FMLP-Rec (WWW'22) | 0.0170 | 0.0477 | 0.0764 | 0.0117 | 0.0216 | 0.0288 | 0.0160 |
| | HSD+BERT4Rec (CIKM'22) | <u>0.0339</u> | <u>0.0732</u> | <u>0.1294</u> | <u>0.0178</u> | <u>0.0305</u> | <u>0.0447</u> | <u>0.0218</u> |
| | AC-BERT4Rec (CIKM'23) | 0.0212 | 0.0445 | 0.0877 | 0.0113 | 0.0188 | 0.0320 | 0.0147 |
| | MSDCCL+BERT4Rec | 0.0708* | 0.1270* | 0.2126* | 0.0449* | 0.0631* | 0.0845* | 0.0498* |
| Beauty | DSAN (AAAI'21) | 0.0092 | 0.0152 | 0.0264 | 0.0058 | 0.0077 | 0.0105 | 0.0062 |
| | FMLP-Rec (WWW'22) | 0.0095 | 0.0166 | 0.0284 | 0.0056 | 0.0078 | 0.0107 | 0.0060 |
| | HSD+BERT4Rec (CIKM'22) | <u>0.0261</u> | <u>0.0447</u> | <u>0.0683</u> | <u>0.0147</u> | <u>0.0207</u> | <u>0.0266</u> | <u>0.0151</u> |
| | AC-BERT4Rec (CIKM'23) | 0.0200 | 0.0371 | 0.0609 | 0.0120 | 0.0175 | 0.0235 | 0.0133 |
| | MSDCCL+BERT4Rec | 0.0522* | 0.0714* | 0.0955* | 0.0378* | 0.0439* | 0.0500* | 0.0372* |
| Sports | DSAN (AAAI'21) | 0.0061 | 0.0105 | 0.0215 | 0.0042 | 0.0056 | 0.0084 | 0.0049 |
| | FMLP-Rec (WWW'22) | 0.0068 | 0.0117 | 0.0180 | 0.0044 | 0.0059 | 0.0075 | 0.0046 |
| | HSD+BERT4Rec (CIKM'22) | <u>0.0120</u> | <u>0.0190</u> | <u>0.0303</u> | <u>0.0078</u> | <u>0.0100</u> | <u>0.0129</u> | <u>0.0081</u> |
| | AC-BERT4Rec (CIKM'23) | 0.0112 | <u>0.0203</u> | <u>0.0351</u> | 0.0069 | 0.0099 | <u>0.0136</u> | 0.0077 |
| | MSDCCL+BERT4Rec | 0.0271* | 0.0387* | 0.0553* | 0.0192* | 0.0229* | 0.0271* | 0.0192* |
| Yelp | DSAN (AAAI'21) | 0.0269 | 0.0369 | 0.0541 | <u>0.0211</u> | 0.0242 | 0.0285 | <u>0.0216</u> |
| | FMLP-Rec (WWW'22) | 0.0203 | 0.0294 | 0.0436 | 0.0142 | 0.0171 | 0.0207 | 0.0144 |
| | HSD+BERT4Rec (CIKM'22) | 0.0292* | 0.0408 | 0.0593 | 0.0223* | 0.0260* | <u>0.0307</u> | 0.0228* |
| | AC-BERT4Rec (CIKM'23) | <u>0.0286</u> | 0.0445* | 0.0700* | 0.0194 | <u>0.0245</u> | 0.0309* | 0.0202 |
| | MSDCCL+BERT4Rec | <u>0.0243</u> | <u>0.0418</u> | <u>0.0695</u> | 0.0151 | <u>0.0207</u> | 0.0277 | 0.0163 |
| ML-1M | DSAN (AAAI'21) | 0.0098 | 0.0336 | 0.0651 | 0.0048 | 0.0122 | 0.0200 | 0.0081 |
| | FMLP-Rec (WWW'22) | 0.0210 | 0.0449 | 0.0707 | 0.0120 | 0.0199 | 0.0263 | 0.0142 |
| | HSD+BERT4Rec (CIKM'22) | <u>0.0477</u> | <u>0.0886</u> | <u>0.1399</u> | <u>0.0297</u> | <u>0.0429</u> | <u>0.0558</u> | <u>0.0328</u> |
| | AC-BERT4Rec (CIKM'23) | OOM | OOM | OOM | OOM | OOM | OOM | OOM |
| | MSDCCL+BERT4Rec | 0.1351* | 0.2095* | 0.3062* | 0.0873* | 0.1112* | 0.1356* | 0.0881* |

* Denotes statistically significant improvement (measured by a two-sided t-test with $p < 0.05$) over the best baseline.

contrastive learning module is set to 0.5, which is annealed after every 40 batches. All embedding parameters are initialized with a Gaussian distribution. For all baseline methods, we either report their performances in the original papers or optimally tuned on the validation data. The early-stopping training strategy is employed to prevent model overfitting, which stops training when the HR@20 metric decreases continuously over 10 epochs on the validation dataset.

5.2. Overall performance comparison

The results of different models on datasets are shown in Tables 2 and 3. The best results are in bold and the second best results are underlined. We conduct the two-sided t-test with $p < 0.05$ to verify the statistical significance for the performance improvement. From the experimental results, we draw the following conclusions:

- Table 2 shows the performance of different sequential recommendation methods with or without our proposed MSDCCL approach over all datasets. We observe that the base models fused with MSDCCL yield significant improvements in most cases when comparing with their counterparts. The results verify the effectiveness of our proposed method which can extract better user preferences together with alleviating the noise issue. To be specific, the relative improvements of MSDCCL+BERT4Rec over BERT4Rec in terms of the metrics HR@20, NDCG@20, MRR@20 are 133.11%, 168.25%, 217.20% respectively on the dataset ML-100k. Similar relative improvements can also be observed on other datasets.
- Table 3 reports the performance comparison of all denoising approaches on all five datasets. We can see that on the ML-100k, Beauty, Sports and ML-1M datasets, the performance of MSDCCL+BERT4Rec, which employs both the soft and hard denoising strategy, consistently outperforms other denoising models relying solely on the soft denoising strategy (e.g., DSAN, FMLP-Rec and AC-BERT4Rec) or hard denoising strategy (e.g., HSD+BEART4Rec). It is worth noting that the performance of MSDCCL+BERT4Rec is suboptimal on the Yelp dataset. This may be due to the fact that the denoising signals produced by MSDCCL are more difficult to fuse with BERT4Rec than the denoising signals produced by

Table 4

Ablation study of our model, we report HR@20, NDCG@20 and MRR@20 on five datasets. The best results are highlighted in bold, and the second best results are underlined.

| Dataset | Metric | Full | w/o DL | w/o TS | w/o BPR | w/o CL |
|---------|---------|---------------|---------------|---------------|---------------|---------------|
| ML-100k | HR@20 | 0.2126 | 0.1783 | 0.2054 | 0.1991 | <u>0.2079</u> |
| | NDCG@20 | 0.0845 | 0.0678 | 0.0823 | 0.0793 | <u>0.0826</u> |
| | MRR@20 | 0.0498 | 0.0380 | <u>0.0489</u> | 0.0468 | 0.0486 |
| Beauty | HR@20 | <u>0.0955</u> | 0.0639 | 0.0957 | 0.0937 | 0.0945 |
| | NDCG@20 | 0.0500 | 0.0279 | 0.0500 | 0.0491 | <u>0.0492</u> |
| | MRR@20 | 0.0372 | 0.0180 | <u>0.0371</u> | 0.0365 | 0.0364 |
| Sports | HR@20 | 0.0553 | 0.0461 | 0.0548 | <u>0.0550</u> | <u>0.0550</u> |
| | NDCG@20 | 0.0271 | 0.0193 | 0.0264 | <u>0.0268</u> | 0.0267 |
| | MRR@20 | 0.0192 | 0.0120 | 0.0185 | <u>0.0191</u> | 0.0188 |
| Yelp | HR@20 | 0.0695 | <u>0.0689</u> | 0.0687 | <u>0.0689</u> | 0.0671 |
| | NDCG@20 | 0.0277 | <u>0.0276</u> | 0.0274 | 0.0273 | 0.0267 |
| | MRR@20 | 0.0163 | 0.0163 | <u>0.0162</u> | 0.0160 | 0.0157 |
| ML-1M | HR@20 | 0.3062 | 0.2826 | 0.3033 | <u>0.3034</u> | 0.3031 |
| | NDCG@20 | 0.1356 | 0.1249 | 0.1337 | <u>0.1346</u> | 0.1336 |
| | MRR@20 | 0.0881 | 0.0811 | 0.0866 | <u>0.0877</u> | 0.0864 |

other models on certain datasets. In summary, these experimental results demonstrate the superiority of our denoising model over other sequence denoising recommendation models in most cases.

5.3. Ablation study

To investigate the role of each component in our proposed MSDCCL, we perform an ablation study by removing each component from the entire model for comparison. Note that we utilize BERT4Rec as the backbone for all variants. To be specific, we consider the following variants of our method:

- **w/o DL**: It discards the Dual Loss with Hard Signal Denoising module, which is developed to explicitly eliminate noisy items from the original sequence with hard signals.
- **w/o TS**: It denotes the variant which drops the target signal from the Target-Aware User Interest Extractor module.

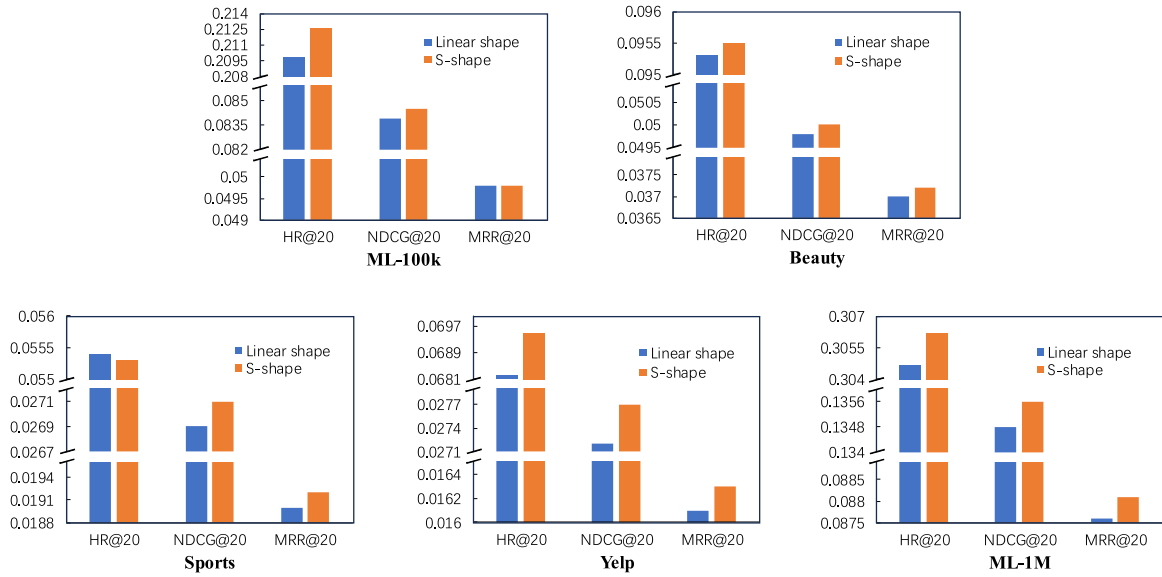


Fig. 4. Analysis on the effectiveness of different increment patterns in curriculum learning, specifically linear shape versus S-shape.

- **w/o BPR**: This variant ignores the supervised loss (i.e., BPR loss) in the Dual Loss with Hard Signal Denoising module.
- **w/o CL**: It removes the curriculum learning module from our model, which indicates that the model will be trained without considering the difficulty of samples.
- **Full**: This model indicates our full model without dropping any component.

As shown in Table 4, we observe that discarding each component of our model MSDCCL will lead to a performance drop in most cases. Specifically, removing the Dual Loss with Hard Signal Denoising module (i.e., w/o DL) will cause a considerable performance decline on all datasets. Compared to MSDCCL, the relative performance decrements of w/o DL are 16.13%, 19.76%, 23.69% in terms of the metrics HR@20, NDCG@20, MRR@20 on the dataset ML-100k. Similar results can be observed on the other datasets. This reveals that the Dual Loss with Hard Signal Denoising module is indeed critical for alleviating the impact of noisy items. Besides, removing the target signal from the Target-Aware User Interest Extractor module (i.e., w/o TS) will also result in an inferior performance, which demonstrates the usefulness of taking the target signal for learning better representations of users. It is interesting to notice that when we remove the supervised loss in the Dual Loss with Hard Signal Denoising module (i.e., w/o BPR), the drop of the performance is less than that of w/o DL. This indicates that both unsupervised loss (i.e., cross-signal contrastive learning loss) and supervised loss (i.e., BPR loss) are necessary in our model. At last, if we exclude the curriculum learning module, the performance will decrease consistently, which indicates the importance of incorporating the extended curriculum learning in our proposed model.

5.4. Effect of S-shape curriculum learning

In this section, we further investigate the effectiveness of developing the S-shape curriculum learning. To the end, we compare the performance of two different increment strategies of training samples (i.e., linear shape increment and S-shape increment) during the curriculum learning process. From Fig. 4, we can see that MSDCCL equipped with S-shape increment performs better than its counterpart equipped with linear increment. This indicates that aligning the increment of training samples with the learning pattern of human beings can provide better learning capacity of curriculum learning in sequential recommendation.

5.5. Comparison with different sequence length

The sequence length reflects the sparsity of user interaction behaviors. To evaluate the performance of our model trained with different sequence lengths, we evenly split sequences of a dataset (e.g., ML-100k and Beauty) into three groups (i.e., short, medium and long) based on their lengths. We compare the performance of our model with the two best performing baselines, including AC-BERT4Rec and HSD+BERT4Rec. According to the results demonstrated in Fig. 5, we see that AC-BERT4Rec obtains superior performance on short sequences as compared with that on both medium and long sequences. This is because these medium and long sequences would contain more noisy items, and AC-BERT4Rec shows inferior results due to the assignment of attention weights to them. On the contrary, the performance of HSD+BERT4Rec on long sequences is much higher than that on both short and medium sequences. This is due to that HSD+BERT4Rec takes a hard denoising strategy which directly discards noisy items in sequences. Since there is no supervised signal for sequence denoising, this method would inevitably overlook relevant items and the problem becomes more serious when the sequence is short. Our model is consistently superior to both AC-BERT4Rec and HSD+BERT4Rec on all three groups. The main reason is that our model can make full use of advantages of both soft and hard denoising strategies where each of them is leveraged to guide the learning process of each other.

5.6. Impact of training set proportion

To investigate the effectiveness of our proposed model with different training set ratios, we compare the performance of our model with the best performing baseline HSD+BERT4Rec on the dataset ML-100k and Beauty. Fig. 6 shows the results when we vary the training set proportion from 20% to 100% with a step size of 20%. It can be seen from the results that the performance of both methods increase gradually with the growth of training data. In addition, our model significantly outperforms HSD+BERT4Rec under all training set ratios, which demonstrates the effectiveness of our model in real-world applications.

5.7. Impact of the Gumbel-softmax strategy

To validate the effectiveness of incorporating the Gumbel-softmax strategy in our proposed method, we replace it with a simple clipping

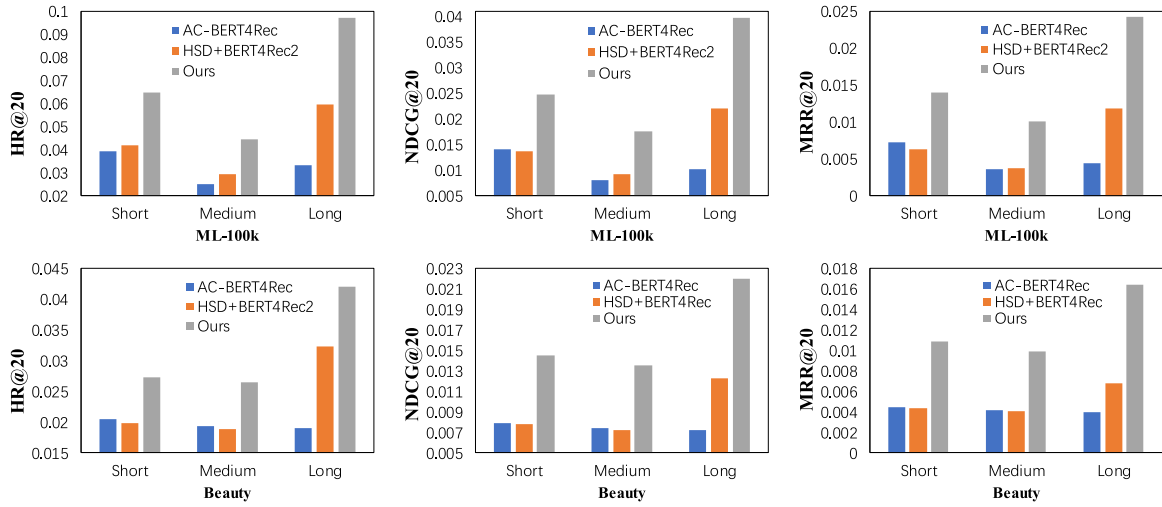


Fig. 5. Performance of our model and the two best performing baselines (i.e., AC-BERT4Rec, HSD+BERT4Rec) under different sequence lengths on ML-100k and Beauty.

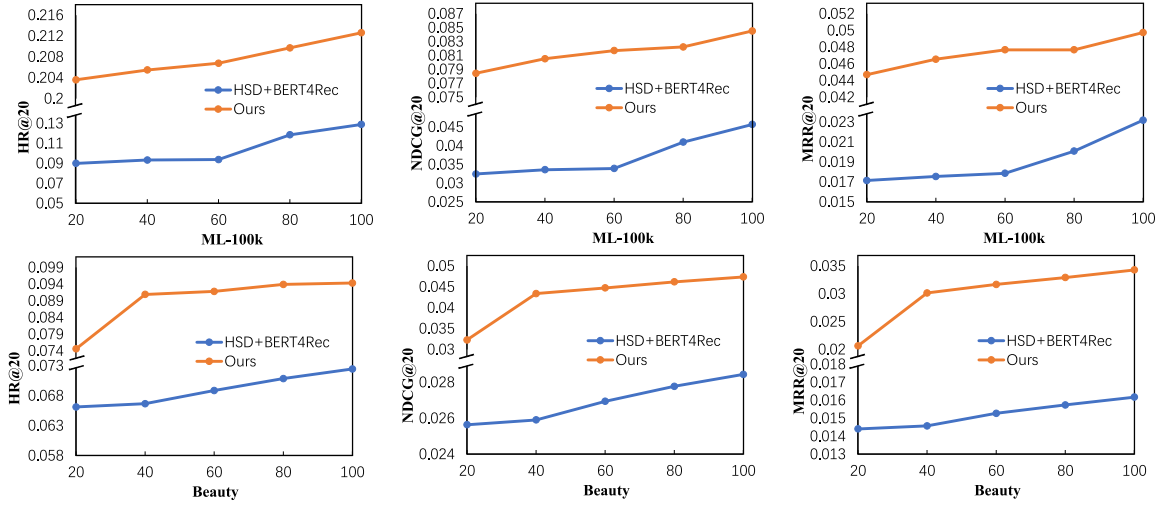


Fig. 6. Performance of our model and the best performing baseline HSD+BERT4Rec with different training set proportion on ML-100k and Beauty.

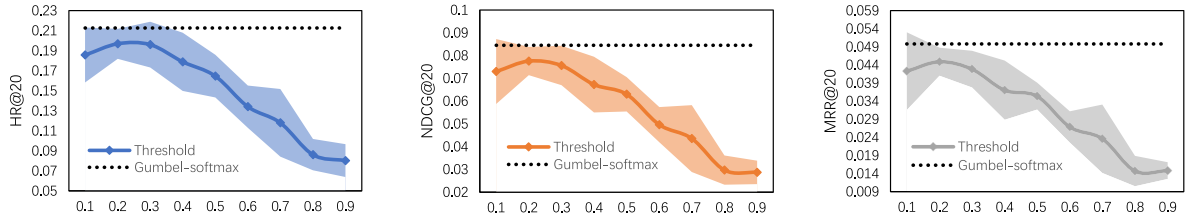


Fig. 7. Performance Comparison of our model with the Gumbel-softmax strategy and the clipping strategy on the ML-100k dataset.

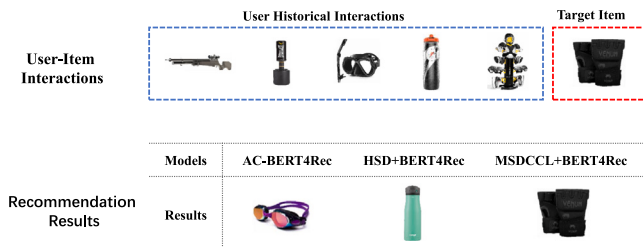


Fig. 8. A case study on Sports to demonstrate the effectiveness of our model.

strategy. We vary the value of the clipping threshold from 0.1 to 0.9 with a step size of 0.1, and the performance on the ML-100k dataset is demonstrated in Fig. 7. We can observe that the model performance with the clipping strategy improves if a higher threshold value is used and reaches a peak when the threshold value is 0.2. Then the performance starts to drop continuously as the threshold value increases. In addition, we can clearly observe that the model performance with the Gumbel-softmax strategy is consistently superior to that of the clipping strategy, which suggests that it is beneficial to employ the Gumbel-softmax strategy in our proposed method. This is probably because

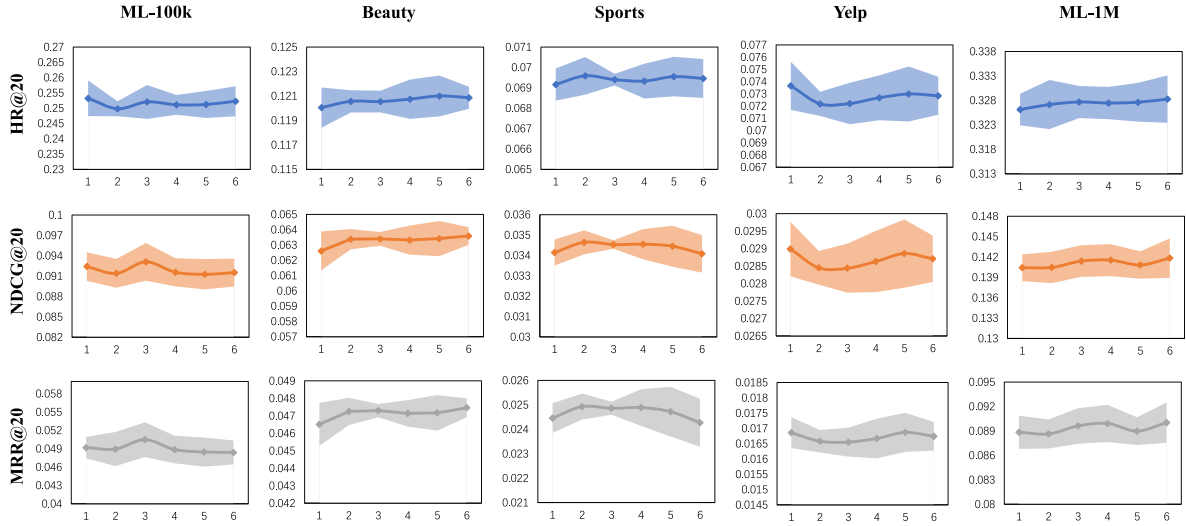


Fig. 9. Performance (HR@20, NDCG@20 and MRR@20) comparison w.r.t hyperparameters m on five datasets.

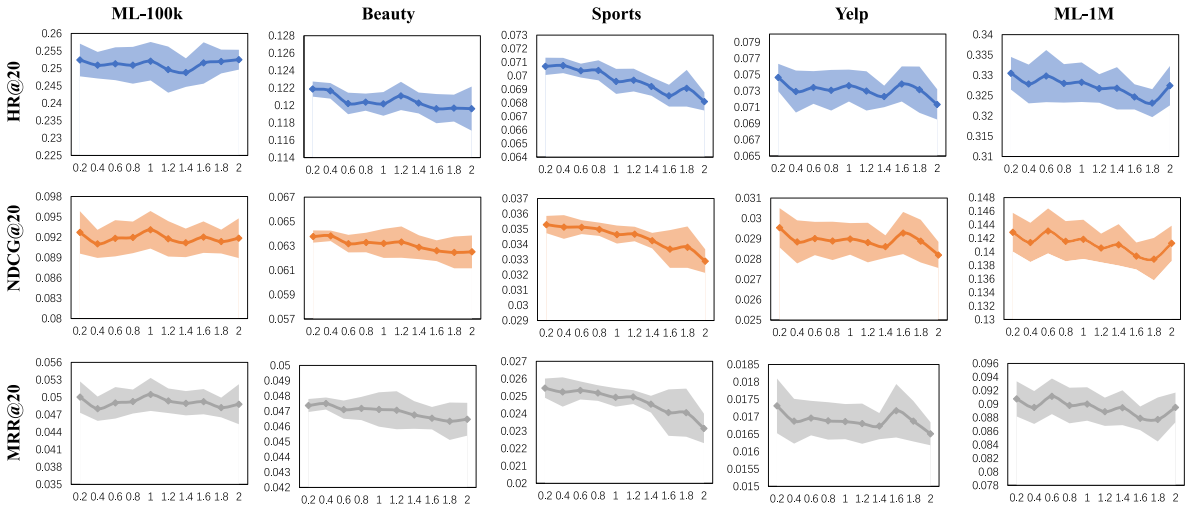


Fig. 10. Performance (HR@20, NDCG@20 and MRR@20) comparison w.r.t hyperparameters λ on five datasets.

the clipping strategy always obtain the same output while the Gumbel-softmax strategy might get probability output due to the incorporation of the Gumbel noise.

5.8. Case study

We further conduct a case study to investigate the advantages of MSDCCL in comparison other denoising models (i.e., soft and hard denoising methods). Fig. 8 shows a historical interaction sequence (in the blue dotted box) and the target item (in the red dotted box) of an indoor sport enthusiast. Among all methods, only the proposed method makes the correct recommendation. The main reason is that solely employing the soft denoising strategy (e.g., AC-BERT4Rec) would still assign low attention weights to noise items such as “gun” and thus affect the recommending performance. In contrast, utilizing the hard denoising strategy (e.g., HSD+BERT4Rec) would discard informative information due to the lack of supervised signal for sequence denoising. Different from these methods, our model can comprehensively leverage the advantages of both soft and hard denoising strategies, thus effectively boosting the capability of sequence denoising and accurately capturing user intent for recommendation.

5.9. Hyperparameter sensitivity

In this section, we investigate the impact of hyperparameters in MSDCCL. We first explore the effect of m , which plays a critical role in modeling short user interests. Then, we analyze the impact of λ which is introduced to balance the importance between the dual loss with hard signal denoising and the recommendation loss. To ensure the experiment is controlled, we change one hyperparameter at one time, while fixing other hyperparameters.

- **Impact of the parameter m .** The parameter m indicates the number of items that are truncated from the end of the sequence. To exploit the impact of the parameter m , we vary it in the range of $\{1, 2, 3, 4, 5, 6\}$ and the performance of MSDCCL is shown in Fig. 9. It can be observed that on all datasets except Yelp, the model’s performance improves with increasing value of m and reaches a peak when m equals to 2 or 3. This phenomenon occurs because the truncated subsequence contains more enriched short-term interest information of the user compared to the last item in the sequence. If we continue to raise the value of m , the performance will start to drop. The reason is that when m becomes too large, more outdated items will be considered to represent user preference, which will inevitably leads to inferior performance.

On the Yelp dataset, we observe fluctuations in the model's performance as m increases. This variability may be due to the presence of noise (i.e., misclicks, malicious false interactions) within the truncated subsequence or sudden shifts in the user's short-term interests, resulting in a suboptimal representation of the user's short-term preferences.

- **Impact of the parameter λ .** We investigate the effect of the hyperparameter λ in Eq. (23), and vary it from 0.2 to 2 with a step size of 0.2. The results are reported in Fig. 10. It can be observed that for all datasets, there is a general decrease in model performance with the increasing value of λ . This observation highlights that it will degrade the performance of sequential recommendations when the Dual Loss with Hard Signal Denoising module takes too much weights in the learning process. It is also worth noting that if we set λ to 0, which is equivalent to discard the Dual Loss with Hard Signal Denoising module from MSDCCL, the performance will drop significantly.

6. Conclusion

In this paper, we propose a novel denoising framework MSDCCL for sequential recommendation. To be specific, we first employ both the soft and hard denoising strategy to alleviate the influence of noisy items in sequences. Then, we capture both user long-term and short-term interest via developing a target-aware user interest extractor. Next, we extend existing curriculum learning by simulating the learning pattern of human beings by utilizing the S-shape increment rather than the conventional linear increment. Extensive experiments on five widely used datasets are leveraged for evaluating the performance of the proposed approach. The results show that MSDCCL can significantly boost the performance of existing mainstream sequential recommendation methods. In addition, MSDCCL is also superior to all state-of-the-art denoising models. For future work, we will propose to employ data augmentation for addressing the denoising issue. To the end, we introduce an auxiliary task which can provide explicit noisy signals for learning robust denoising models. In addition, we also attempt to investigate other patterns, instead of S-shape increment, for increasing “difficult” instances in curriculum learning.

CRedit authorship contribution statement

Xiaofei Zhu: Writing – review & editing, Writing – original draft, Supervision, Methodology, Conceptualization. **Liang Li:** Writing – original draft, Software, Methodology, Conceptualization. **Weidong Liu:** Supervision. **Xin Luo:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [grant number 62141201]; the Natural Science Foundation of Chongqing, China [grant number CSTB2022NSQ-MSX1672]; the Chongqing Talent Plan Project, China [grant number CSTC2024 YCJH-BGZXMM0022]; the Major Project of Science and Technology Research Program of Chongqing Education Commission of China [grant number KJZD-M202201102].

References

- Cai, R., Wu, J., San, A., Wang, C., & Wang, H. (2021). Category-aware collaborative sequential recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 388–397).
- Chaudhary, L., & Singh, B. (2023). Gumbel-SoftMax based graph convolution network approach for community detection. *International Journal of Information Technology*, 15(6), 3063–3070.
- Chen, H., Lin, Y., Pan, M., Wang, L., Yeh, C. M., Li, X., et al. (2022). Denoising self-attentive sequential recommendation. In *Proceedings of the 16th ACM conference on recommender systems* (pp. 92–101).
- Chen, Y., Liu, Z., Li, J., McAuley, J. J., & Xiong, C. (2022). Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM web conference 2022* (pp. 2172–2182).
- Chen, X., Wang, Z., Xu, H., Zhang, J., Zhang, Y., Zhao, W. X., et al. (2023). Data augmented sequential recommendation based on counterfactual thinking. *IEEE Transactions on Knowledge and Data Engineering*, 35(9), 9181–9194.
- Duan, C., Fan, W., Zhou, W., Liu, H., & Wen, J. (2023). Clsprec: Contrastive learning of long and short-term preferences for next POI recommendation. In *Proceedings of the 32nd ACM international conference on information and knowledge management* (pp. 473–482).
- Fu, T., & Chen, J. (2023). Curriculum learning for debiased recommendation with explicit and implicit feedback. *Journal of Physics: Conference Series*, 2504(1), Article 012052.
- Gonzalez-Toral, R., Reviriego, P., Maestro, J. A., & Gao, Z. (2018). A scheme to design concurrent error detection techniques for the fast Fourier transform implemented in SRAM-based FPGAs. *IEEE Transactions on Computers*, 67(7), 1039–1045.
- Hao, Y., Ma, J., Zhao, P., Liu, G., Xian, X., Zhao, L., et al. (2023). Multi-dimensional graph neural network for sequential recommendation. *Pattern Recognition*, 139, Article 109504.
- Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2016). Session-based recommendations with recurrent neural networks. In *Proceedings of the 4th international conference on learning representations*.
- Huang, C., Wang, X., He, X., & Yin, D. (2022). Self-supervised learning for recommender system. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 3440–3443).
- Kang, W.-C., & McAuley, J. (2018). Self-attentive sequential recommendation. In *Proceedings of the 18th international conference on data mining* (pp. 197–206).
- Krichene, W., & Rendle, S. (2020). On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1748–1757).
- Li, Y., Chen, H., Sun, X., Sun, Z., Li, L., Cui, L., et al. (2021). Hyperbolic hypergraphs for sequential recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 988–997).
- Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., & Ma, J. (2017). Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on conference on information and knowledge management* (pp. 1419–1428).
- Li, J., Wang, Y., & McAuley, J. (2020). Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining* (pp. 322–330).
- Lin, Q., Zhou, W. J., Wang, Y., Da, Q., Chen, Q.-G., & Wang, B. (2022). Sparse attentive memory network for click-through rate prediction with long sequences. In *Proceedings of the 31st ACM international conference on information & knowledge management* (pp. 3312–3321).
- Liu, Q., Wu, S., Wang, D., Li, Z., & Wang, L. (2016). Context-aware sequential recommendation. In *Proceedings of the 16th international conference on data mining* (pp. 1053–1058).
- Liu, Q., Zeng, Y., Mokhosi, R., & Zhang, H. (2018). STAMP: Short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1831–1839).
- Luo, A., Zhao, P., Liu, Y., Zhuang, F., Wang, D., Xu, J., et al. (2020). Collaborative self-attention network for session-based recommendation. In *Proceedings of the 29th international joint conference on artificial intelligence* (pp. 2591–2597).
- Lv, Y., Zhuang, L., Luo, P., Li, H., & Zha, Z. (2020). Time-sensitive collaborative interest aware model for session-based recommendation. In *2020 IEEE international conference on multimedia and expo* (pp. 1–6).
- Ma, M., Ren, P., Chen, Z., Ren, Z., Liang, H., Ma, J., et al. (2023). Improving transformer-based sequential recommenders through preference editing. *ACM Transactions on Information Systems*, 41(3), 71:1–71:24.
- Murre, J. M. (2014). S-shaped learning curves. *Psychonomic Bulletin & Review*, 21, 344–356.
- Orvieto, A., Smith, S. L., Gu, A., Fernando, A., Gülçehre, Ç., Pascanu, R., et al. (2023). Resurrecting recurrent neural networks for long sequences. In *Proceedings of the 15th international conference on machine learning: vol. 202*, (pp. 26670–26698).
- Qin, Y., Ju, W., Wu, H., Luo, X., & Zhang, M. (2024). Learning graph ODE for continuous-time sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

- Qiu, R., Huang, Z., & Yin, H. (2021). Memory augmented multi-instance contrastive predictive coding for sequential recommendation. In *Proceedings of the 2021 IEEE international conference on data mining* (pp. 519–528).
- Qiu, R., Huang, Z., Yin, H., & Wang, Z. (2022). Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the 15th ACM international conference on web search and data mining* (pp. 813–823).
- Quan, Y., Ding, J., Gao, C., Yi, L., Jin, D., & Li, Y. (2023). Robust preference-guided denoising for graph based social recommendation. In *Proceedings of the ACM web conference 2023* (pp. 1097–1108).
- Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th conference on uncertainty in artificial intelligence* (pp. 452–461).
- Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L. (2010). Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on world wide web* (pp. 811–820).
- Sudarsan, V. J., & Polash, M. M. A. (2023). Self-attention convolutional neural network for sequential recommendation. In *Proceedings of the 24th international conference: vol. 14306*, (pp. 569–578).
- Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., et al. (2019). BERT4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1441–1450).
- Sun, Y., Wang, B., Sun, Z., & Yang, X. (2021). Does every data instance matter? Enhancing sequential recommendation by eliminating unreliable data. In *Proceedings of the 30th international joint conference on artificial intelligence* (pp. 1579–1585).
- Sun, Y., Yuan, F., Yang, M., Wei, G., Zhao, Z., & Liu, D. (2020). A generic network compression framework for sequential recommender systems. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 1299–1308).
- Tang, J., & Wang, K. (2018). Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining* (pp. 565–573).
- Tolomei, G., Lalmas, M., Farahat, A., & Haines, A. (2019). You must have clicked on this ad by mistake! Data-driven identification of accidental clicks on mobile ads with applications to advertiser cost discounting and click-through rate prediction. *International Journal of Data Science and Analytics*, 7, 53–66.
- Tong, X., Wang, P., Li, C., Xia, L., & Niu, S. (2021). Pattern-enhanced contrastive policy learning network for sequential recommendation. In *Proceedings of the 30th international joint conference on artificial intelligence* (pp. 1593–1599).
- Wang, W., Feng, F., He, X., Nie, L., & Chua, T. S. (2021). Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining* (pp. 373–381).
- Wang, Z., Gao, M., Li, W., Yu, J., Guo, L., & Yin, H. (2023). Efficient bi-level optimization for recommendation denoising. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 2502–2511).
- Wang, X., Jia, L., Guo, L., & Liu, F. (2023). Multi-aspect heterogeneous information network for MOOC knowledge concept recommendation. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, 53(10), 11951–11965.
- Wang, L., Lim, E., Liu, Z., & Zhao, T. (2022). Explanation guided contrastive learning for sequential recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management* (pp. 2017–2027).
- Wang, Y., Liu, Y., Wang, Q., Wang, C., & Li, C. (2023). Poisoning self-supervised learning based sequential recommendations. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval* (pp. 300–310).
- Wang, X., Pan, Z., Zhou, Y., Chen, H., Ge, C., & Zhu, W. (2023). Curriculum co-disentangled representation learning across multiple environments for social recommendation. In *Proceedings of the 40th international conference on machine learning* (pp. 36174–36192).
- Wang, C., Zhang, M., Ma, W., Liu, Y., & Ma, S. (2020). Make it a chorus: Knowledge- and time-aware item modeling for sequential recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 109–118).
- Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., & Tan, T. (2019). Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence: vol. 33*, (no. 01), (pp. 346–353).
- Wu, J., Wang, X., Feng, F., He, X., Chen, L., Lian, J., et al. (2021). Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 726–735).
- Xia, X., Yin, H., Yu, J., Wang, Q., Cui, L., & Zhang, X. (2021). Self-supervised hyper-graph convolutional networks for session-based recommendation. In *Proceedings of the 35th AAAI conference on artificial intelligence* (pp. 4503–4511).
- Xie, X., Sun, F., Liu, Z., Wu, S., Gao, J., Zhang, J., et al. (2022). Contrastive learning for sequential recommendation. In *Proceedings of the 38th international conference on data engineering* (pp. 1259–1273).
- Xu, H., Peng, Q., Liu, H., Sun, Y., & Wang, W. (2024). Group-based personalized news recommendation with long- and short-term fine-grained matching. *ACM Transactions on Information Systems*, 42(1), 3:1–3:27.
- Xu, X., Yuan, B., Mo, Y., Song, T., & Li, S. (2024). Curriculum recommendations using transformer base model with InfoNCE loss and language switching method. *arXiv preprint arXiv:2401.09699*.
- Xu, C., Zhao, P., Liu, Y., Sheng, V. S., Xu, J., Zhuang, F., et al. (2019). Graph contextualized self-attention network for session-based recommendation. In *Proceedings of the 28th international joint conference on artificial intelligence: vol. 19*, (pp. 3940–3946).
- Yu, J., Yin, H., Xia, X., Chen, T., Cui, L., & Nguyen, Q. V. H. (2022). 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* (pp. 1294–1303).
- Yuan, F., Karatzoglou, A., Arapakis, I., Jose, J. M., & He, X. (2019). A simple convolutional generative network for next item recommendation. In *Proceedings of the twelfth ACM international conference on web search and data mining* (pp. 582–590).
- Yuan, J., Song, Z., Sun, M., Wang, X., & Zhao, W. X. (2021). Dual sparse attention network for session-based recommendation. In *Proceedings of the AAAI conference on artificial intelligence: vol. 35*, (no. 5), (pp. 4635–4643).
- Zhang, C., Chen, R., Zhao, X., Han, Q., & Li, L. (2023). Denoising and prompt-tuning for multi-behavior recommendation. In *Proceedings of the ACM web conference 2023* (pp. 1355–1363).
- Zhang, C., Du, Y., Zhao, X., Han, Q., Chen, R., & Li, L. (2022). Hierarchical item inconsistency signal learning for sequence denoising in sequential recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management* (pp. 2508–2518).
- Zhang, H., Li, Y., Ding, B., & Gao, J. (2020). Practical data poisoning attack against next-item recommendation. In *Proceedings of the web conference 2020* (pp. 2458–2464).
- Zheng, Y., Gao, C., Chang, J., Niu, Y., Song, Y., Jin, D., et al. (2022). Disentangling long and short-term interests for recommendation. In *Proceedings of the ACM web conference 2022* (pp. 2256–2267).
- Zheng, Y., Liu, S., Li, Z., & Wu, S. (2021). Cold-start sequential recommendation via meta learner. In *Proceedings of the AAAI conference on artificial intelligence: vol. 35*, (no. 5), (pp. 4706–4713).
- Zhou, K., Wang, H., Zhao, W. X., Zhu, Y., Wang, S., Zhang, F., et al. (2020). S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 1893–1902).
- Zhou, P., Ye, Q., Xie, Y., Gao, J., Wang, S., Kim, J. B., et al. (2023). Attention calibration for transformer-based sequential recommendation. In *Proceedings of the 32nd ACM international conference on information and knowledge management* (pp. 3595–3605).
- Zhou, K., Yu, H., Zhao, W. X., & Wen, J. R. (2022). Filter-enhanced MLP is all you need for sequential recommendation. In *Proceedings of the ACM web conference 2022* (pp. 2388–2399).
- Zhu, Z., Li, S., Liu, Y., Zhang, X., Feng, Z., & Hou, Y. (2024). High-level preferences as positive examples in contrastive learning for multi-interest sequential recommendation. *World Wide Web*, 27(2), 21.