Front. Comput. Sci., 2024, 18(6): 186333
https://doi.org/10.1007/s11704-023-3112-y

RESEARCH ARTICLE

A general tail item representation enhancement framework for sequential recommendation

Mingyue CHENG[1], Qi LIU (✉)[1], Wenyu ZHANG[1], Zhiding LIU[1], Hongke ZHAO[2], Enhong CHEN[1]

1 Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence, Hefei 230026, China
2 College of Management and Economics, Tianjin University, Tianjin 300072, China

Abstract Recently advancements in deep learning models have significantly facilitated the development of sequential recommender systems (SRS). However, the current deep model structures are limited in their ability to learn high-quality embeddings with insufficient data. Meanwhile, highly skewed long-tail distribution is very common in recommender systems. Therefore, in this paper, we focus on enhancing the representation of tail items to improve sequential recommendation performance. Through empirical studies on benchmarks, we surprisingly observe that both the ranking performance and training procedure are greatly hindered by the poorly optimized tail item embeddings. To address this issue, we propose a sequential recommendation framework named *TailRec* that enables contextual information of tail item well-leveraged and greatly improves its corresponding representation. Given the characteristics of the sequential recommendation task, the surrounding interaction records of each tail item are regarded as contextual information without leveraging any additional side information. This approach allows for the mining of contextual information from cross-sequence behaviors to boost the performance of sequential recommendations. Such a light contextual filtering component is plug-and-play for a series of SRS models. To verify the effectiveness of the proposed *TailRec*, we conduct extensive experiments over several popular benchmark recommenders. The experimental results demonstrate that *TailRec* can greatly improve the recommendation results and speed up the training process. The codes of our methods have been available [1)].

Keywords sequential recommendation, long-tail distribution, training accelerating

1 Introduction

Sequential recommender systems (SRS) are increasingly prevalent due to their ability to infer dynamic user preferences better than traditional collaborative filtering methods. Recent literature [1−3] has shown that deep learning models have facilitated sequential recommendation performance improvements. Most existing deep SRS models are typically constructed based on a sandwich-structured deep neural network, involving three major modules: two embedding layers for input and output items, and several hidden layers [4−6]. Recent advancements have demonstrated promising results with larger model sizes and more model parameters [7,8]. The performance of the deep SRS architecture heavily depends on the scale and quality of historical behaviors [9]. To fulfill the capacity of these deep SRS models, sufficient historical interactions for each item are necessary to ensure that the representations of items can be sufficiently trained [10−12]. However, the number of interactions for each item exhibits a long-tail distribution [13−15] in real-world applications, i.e., meaning that most items only have low-frequency interactions.

Assuming that recommendation models trained on highly skewed distributed data are prone to highlighting popular items, a large body of research have been proposed to eliminate the popularity bias on general recommendation tasks [14]. Roughly, previous strategies can be roughly classified into two categories: 1) modifying the loss function [16], and 2) leveraging more side information [11]. The former idea is to re-weight the original loss or add regularization terms so as to impact the training process of recommendation models. The latter idea is to further leverage content features as additional information to enrich the representation of tail items. Likewise, some methods [11,17,18] have also been proposed in the context of sequential recommendation by taking the diversity results into account. Despite their effectiveness, few efforts are devoted to focusing on studying the representation quality of sequential recommender trained using such long-tail distributed data. Besides the popularity bias, one may wonder

Received February 7, 2023; accepted June 27, 2023

E-mail: qiliuql@ustc.edu.cn

1) See github.com/Mingyue-Cheng/TailRec website.

whether the skewed distributed data would take more unseen influence beyond the current observed findings in the recommendation area on benchmark recommenders.

With these analyses in mind, in this work, we aim to study the effects of long-tail distribution on sequential recommendation by focusing on the representation quality of tail items. The representation quality of tail items is vital in inferring each user's long- and short-term interests. Current sequential recommenders typically adopt the back-propagation mechanism to train suitable embedding for each item. High-quality embeddings can be learned for popular items classified by global interaction frequency with the help of sufficient supervision signals. However, the contextual descriptions of tail items might fail to accurately characterize the raw items, leading to poorly optimized embeddings that can be regarded as noisy inputs. This can hinder the SRS from effectively understanding the user's true preference over candidate items. To validate our thoughts, we conduct several in-depth empirical studies by testing prevalent benchmark recommenders [19]. As illustrated in Fig. 1, the accuracy of the whole SRS is heavily dragged down by the poorly optimized representation of tail items. To deeply understand the reason for performance degradation, we also plot the curves of the learning process in Fig. 2. Surprisingly, we observe that deep SRS architecture typically needs to spend more time converging subject to the noisy input of long-tail data. That is, the inadequate contextual descriptions of tail items might heavily slow down the training process of recommender architectures. From an empirical view, these findings show the evidence that the poorly optimized tail item embeddings in sequential recommender significantly hinder both the final recommendation performance and the training efficiency.

To solve the issues above, we propose an easily compatible sequential recommendation framework named *TailRec*, which incorporates a plug-and-play contextual representation module. The core idea of *TailRec* is to leverage contextual information of tail items to further improve their representation quality. The surrounding interaction records of each tail item are considered as contextual information without using any additional side information. Since co-occurring items are typically consumed by users with similar interests, this setting makes sense. Each tail item has different contextual information because any item can be consumed by multiple users. Therefore, we allow the contextual representation module to dynamically update during training to fully utilize this co-occurrence information. Notably, cross-sequence information can be effectively mined in sequential recommendation tasks. The representation quality of tail items can be significantly improved by incorporating contextual information, which enhances both the recommendation results and training efficiency. We evaluate *TailRec* over three benchmark recommender models for sequential recommendation tasks and demonstrate its effectiveness and efficiency. The experimental results consistently show that the benchmark sequential recommenders can benefit a lot from the newly incorporated contextual representation module. In summary, this work makes the following contributions:

- We present empirical evidence that poorly represented tail items not only significantly harm sequential recommendation performance but also impede the training process. This justifies the importance of our research questions.
- We propose a general sequential recommendation framework named *TailRec* to focus on improving the embedding quality of tail items. In *TailRec*, the newly designed contextual representation module is very easy to leverage but also applicable for a series of sequential recommenders.
- We conduct experiments to validate the effectiveness of *TailRec* on multiple benchmark recommenders. The experimental results demonstrate that *TailRec* can not only produces more promising recommendation performances than baselines but also significantly accelerate the training process of benchmark recommender models.

2 Preliminaries

In this section, we will first formulate the sequential recommendation task. Then, we will briefly revisit prevalent sequential recommenders and identify poorly learned embeddings of tail items as the bottleneck of current deep SRS models. Subsequently, we will provide in-depth empirical studies to support our arguments. The novel contribution of this section is to uncover two vital under-explored findings for deep SRS models caused by the long-tail distributed data.

2.1 Problem formulation

Let $\mathcal{U}$ be the set of M unique users, and $\mathcal{I}$ be the set of N unique items. Using global interaction frequency, we divide the item set into short-head (popular) items $\mathcal{I}^{head}$ and tail

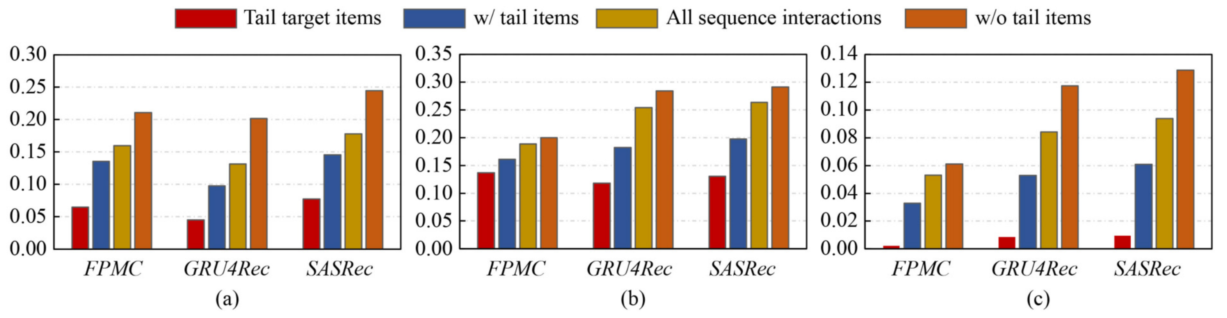

Fig. 1 Evaluation results of NDCG@10 on three benchmark recommenders over three public datasets. (a) Evaluation on MovieLens; (b) evaluation on Yoochoose; (c) evaluation on Deginetica

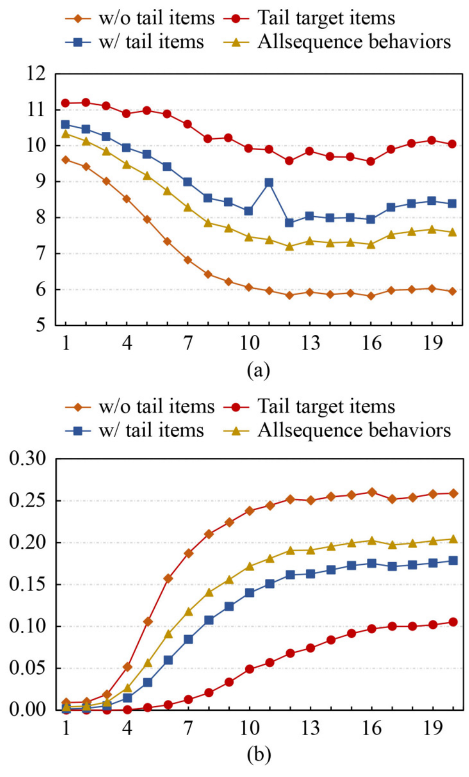

Fig. 2 Testing loss and NDCG of *SASRec* on Deginetica. (a) Testing loss on *SASRec*; (b) Testing NDCG on *SASRec*

items $\mathcal{I}^{tail}$. Each user is represented by a sequence of items, denoted by $X^u = x_1^u, x_2^u, \ldots, x_t^u$, where x_t^u represents the tth item the user interacted with, ordered by time. Given this historical sequential behavior, the task of sequential recommendation systems (SRS) is to predict the item x_{t+1}^u that the user is likely to interact with at time $t+1$, which can be achieved by modeling the probability of all candidate items $P(i_{t+1} = i | i_{1:t})$. In practice, SRS usually recommends more than one item by selecting the top-K items from $\mathcal{I}$.

2.2 Empirical studies w.r.t. tail items for SRS

To investigate the impact of tail items, we conduct a comprehensive empirical study by testing benchmark recommenders. Specifically, we evaluate the recommendation performance of three widely used sequential recommendation methods (FMPC [20], GRU4Rec [21], and SASRec [19]) across three benchmark datasets. Additionally, we establish four different evaluation scopes to measure the effects of tail items on SRS.

- **tail target items** means that only tail items were included in the sequence to be predicted, i.e., target items.
- **w/ tail items** denotes that at least one tail item can be found in the selected sequence behaviors to be evaluated. Note that the tail item in this context may occur in either the past interacted sequences or the position of the target item.
- **all sequential interactions** indicates all sequential behaviors are evaluated regardless of whether involving the tail items within current sequence interactions.
- **w/o tail items** indicates that no tail items can be found in evaluated sequential behaviors.

For all reported results, we conduct rigorous controlled experiments. More information about the hyper-parameters and model details can be found in Section 4.1. As illustrated in Fig. 1, we observe a consistent trend across benchmark recommenders, whereby the presence of tail items led to performance degradation for all recommendation models. Specifically, the accuracy performances of the four evaluation scopes are ranked as: w/o tail item, all sequence interactions, w/ tail items, tail target items. To some extent, this result supports our argument that the learned embedding information is insufficient to characterize raw tail items, which makes it difficult to train these items effectively. We believe that this finding is intuitively reasonable. Representing each user by aggregating all historical interaction behaviors is the default setting in sequential recommendation tasks. However, the poorly optimized embeddings of tail items have a negative impact on the quality of user representations, hindering the model's ability to predict user preferences for candidate items. It also deserves to be noticed that the performance of "tail target items" is significantly lower than several other evaluation scopes. This is mainly probably because of the matching mechanism between user embedding and candidate items.

To gain a deeper understanding of the experimental phenomenon reported in Fig. 1, we plotted the curves of the learning process recorded by testing loss and testing NDCG@10 in SASRec [19]. Lower testing loss reflects better recommendation accuracy and is aligned with the curves of testing NDCG@10. Surprisingly, as shown in Fig. 2, we found that the presence of a large proportion of tail items significantly slows down the entire training process. We hypothesize that the poorly optimized embedding of tail items largely hinders the learning process of the whole sequence dependence. Thus, it is crucial to address the issue of inefficient data utilization caused by inaccurate embeddings of tail items to improve the performance of SRS models. However, one may wonder whether the performance degradation is caused by the popularity bias [16], which suggests that recommendation models are prone to bias towards popular items. To some extent, we also agree with that there might be a connection between the popularity bias and unsatisfactory recommendation results. We leave this for future work. In this study, our main aim is to improve the recommendation results and accelerate the training process by resolving the noisy input representation of tail items.

3 Methodology

Motivated by these empirical findings, we set the goal of this work to solve the ineffectiveness and inefficiency issues of deep SRS incurred by poorly optimized embeddings of tail item set. Thus, we would describe the proposed *TailRec* framework in detail. For ease of understanding, we describe *TailRec* on the top of a prevalent sequential recommender model [19,22]. In the following, we first briefly introduce the base sequential recommendation backbone network. Then, we describe how we utilize collaborative signals from the co-occurrence popular items to boost the representation of tail items.

3.1 Base sequential recommender

In this subsection, we describe how we model the historical sequential behaviors with self-attentive based sequential recommender architecture by stacking the embedding layer, hidden layers, and prediction layer, in turn.

3.1.1 Embedding layer

In the embedding mapping stage, we initialize an embedding matrix denoted by $\mathbf{M}$ with dimensions $\mathbb{R}^{N\times d}$ to encode each item ID into a latent vector of size d. To obtain the input embedding matrix $\mathbf{E}^u$ with dimensions $\mathbb{R}^{t\times d}$ for a sequence of behaviors with a length of t denoted as X^u, we perform a look-up operation on the embedding matrix $\mathbf{M}$.

To enhance the input representation of the item sequence, we utilize a learnable position embedding matrix, designated as $\mathbf{P}$, with dimensions of $\mathbb{R}^{t\times d}$. The position embedding matrix enables the model to capture the sequential order of the items in the sequence. Finally, to obtain the sequence interaction representation, we sum the input embedding matrix and the position embedding matrix directly, which is represented as $\mathbf{E}^u = \mathbf{E}^u + \mathbf{P}$.

3.1.2 Self-attentive hidden layers

Followed by the embedding layer, we then feed the embeddings of interacted sequence into a stack of hidden residual layers, i.e., Transformer blocks, which are expected to capture the dynamic interests of users. Formally, the Lth residual block is abstracted as below:

$$\begin{aligned}\mathbf{H}^u_L &= f(\mathbf{H}^u_{L-1};\Theta_L),\\ &= \mathrm{LN}(\alpha_{L-1}\mathcal{F}_L(\mathbf{H}^u_{L-1})+\mathbf{H}^u_{L-1}),\end{aligned} \tag{1}$$

in which LN represents the layer normalization operation, while $\mathbf{H}^u_{L-1}$ and $\mathbf{H}^u_L \in \mathbb{R}^{k\times t}$ represent the input and output of the Lth residual block, respectively. Here, $\mathcal{F}L(\mathbf{H}^u_{L-1})$ is the residual mapping to be learned, where k denotes the hidden size. Additionally, Θ_L represents the parameter in the Lth layer. Recently proposed works [23] suggest a slight modification to the original residual layers by allowing the weight α_L to be learnable. Such a simple operation enables the hidden layer to be stacked up to deeper layers. Thanks to this modification, the backbone network can achieve not only fast convergence but also better accuracy.

3.1.3 Prediction layer

After L layers, we obtain the user representation at each time step, which extracts the evolving user interest. Following previous works [19,24], we set the last item representation as the final user representation to represent her preference vectors. Then, we compute the users' preference score for the item i at the time step $t+1$ under the context of sequential behaviors as

$$P(i_{t+1}=i|i_{1:t}) = \mathbf{w_i}^{\top}\cdot\mathbf{H}^L_t, \tag{2}$$

where $\mathbf{w}_i$ denotes the representation of item i in the prediction layer. Here, we regard the cross-entropy loss [25] as the optimization objective to guide the training process of recommender network.

3.2 The *TailRec* framework

In this section, we will start by giving an overview of the newly proposed *TailRec* framework. Following that, we will delve into the contextual representation module, which is a critical component in our proposed TailRec framework.

3.2.1 Overview of the proposed *TailRec*

As depicted in Fig. 3, the proposed *TailRec* framework features an additional contextual representation component, which is designed and maintained during training to efficiently capture contextual information for each tail item. Unlike traditional sequential recommenders, we propose to fuse the contextual information of surrounding consumed items as the tail item's contextual information, without relying on any extra side information of users or items. This approach is based on the assumption that users can be typically represented by the items they interact with [26], while items can also be represented by the users who interact with them [27]. Given that each tail item may have been consumed by multiple users, our contextual representation module is dynamically updated to ensure that it maintains a well-informed contextual representation. This allows for the deep SRS to further improve the utilization of historical interactions data, and learn high-quality embeddings to better characterize user preferences. Overall, the incorporation of this component provides a significant enhancement to our framework and

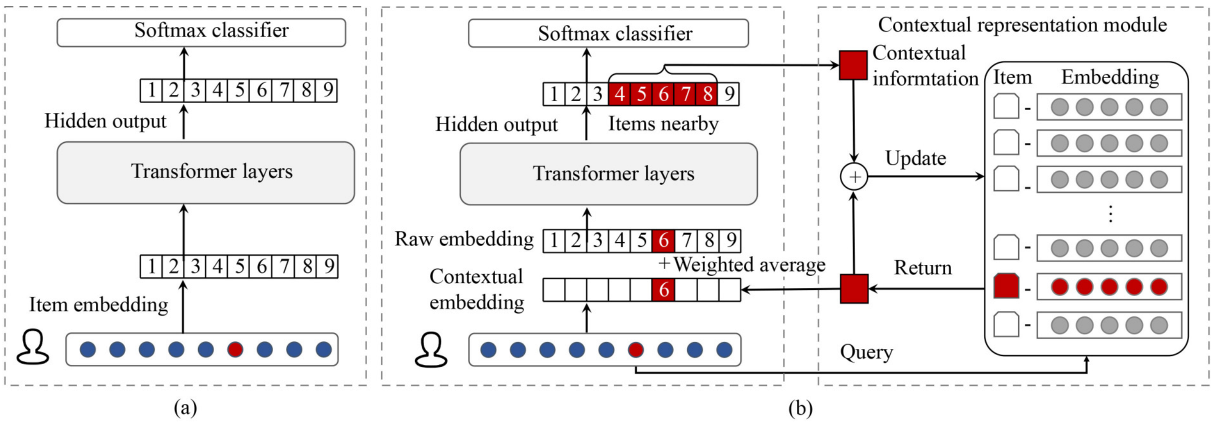

Fig. 3 (a) Base sequential recommender; (b) contextual representation enhanced sequential recommender framework

improves its ability to handle tail items, making it a powerful tool for personalized recommendation systems.

3.2.2 Constructing the contextual representation module

As previously demonstrated, the trained representations of popular items are adequate and can greatly contribute to representing a user's preferences. However, a major drawback of the current deep SRS is that a significant number of less popular items cannot be effectively trained since they occur only a limited number of times. Therefore, we decide to enhance these unpopular items by transferring accurate embedding knowledge from co-occurring items within the same sequential behaviors. Ideally, such contextual representation would cover all tail items, but the lower frequency of the enhanced items may result in inaccurate information from its limited surrounding items. This is due to the insufficient co-occurrence items that are used to extract contextual signals. In this study, we limit the scope of the enhanced tail items to $\mathcal{I}_e^{tail}$, excluding excessively raw tail items. More detailed implementation information can be found in Section 4.1. The enhanced embedding matrix for these tail items is referred to as $\mathbf{E}_e$. To avoid introducing new inductive biases, we use the same initialization manner for this newly enhanced embedding matrix.

3.2.3 Updating the contextual representation module

We are committed to finding useful collaborative signals to update and maintain the enhanced representation component. In recommender systems, users can be typically represented by the interacted items and items also can be represented by the interacted users. Thus, one feasible way to update the representation of the tail item is based on the interacted users' historical interaction behaviors. We proposed to learn the enhanced representation of the target item based on the co-occurrence items of it. Among these co-occurrence items, head items that have high-quality embeddings can provide accurate embeddings as the complementary for tail items. We believe this is reasonable since we all know that items that users have interacted with are out of the same user interest. Thus these items are highly complementary in embeddings. However, each user may only have part of the interest that is aligned with the tail item. From this point, it is not appropriate that treat all co-occurred items to update the collaborative representation module. Hence, we perform the parameter updating with a more flexible form by moving a stride window to extract only local information to construct the relative embedding signals to update the enhanced embedding module. Then, the information extraction for enhanced tail items can be abstracted as

$$\mathbf{E}_c = \frac{1}{2s}\sum_{i=t-s}^{t+s} e_i, \tag{3}$$

where $\mathbf{E}_c$ denote the extracted contextual embeddings, and s denotes the half window size that controls how many surrounding items we want to memorize. In addition, e_i denotes embedding of contextual items.

With the contextual embedding calculated by Eq. (3), we would update the contextual representation module with the latest embedding distilled cross co-occurred sequential behaviors during training. Particularly, we propose to update the contextual representation component using the exponential moving average by following recently proposed work [28]. Formally, such updating process can be described as

$$\mathbf{E}_e = (1-\lambda)\cdot\mathbf{E}_e + \lambda\cdot\mathbf{E}_c, \tag{4}$$

where the $\lambda \in (0,1)$ is the discount trade-off to control the degree of how much the latest enhanced representation would update the contextual representation module. Actually, such weight-average mechanism can be more suitable for updating the enhanced embedding module than gradient descent mechanism. More experimental analysis w.r.t. the impacts of the updating manner would be discussed in Section 4.2.

3.2.4 Leveraging the contextual representation module

By the above design, the contextual representation module can be effectively maintained by leveraging the surrounding items of each tail item. This contextual information can be directly utilized to enhance the learning of deep SRS. Specifically, we incorporate this contextual information into the embedding layer of the entire recommendation network. We achieve this by simply combining the raw representation $\mathbf{E}^u$ with the newly enhanced representation $\mathbf{E}_e^u$ in a weighted average manner. Formally, the fusion process can be described as follows:

$$\mathbf{E}^i = \begin{cases} (1-\gamma)\cdot\mathbf{E}^i + \gamma\cdot\mathbf{E}_e^i, & \text{if item } i \in \mathcal{I}_e^{tail}, \\ \mathbf{E}^i, & \text{if item } i \in \mathcal{I}^{head}, \end{cases}$$

in which the parameter γ is controllable and determines the extent to which we utilize the additional contextual representation in encoding the tail item. Through empirical analysis, we have found that setting γ to 0.9 yields excellent results. This approach enables the model to obtain more meaningful information, which results in accurate embeddings for characterizing tail items.

4 Experiments

We first compare the effectiveness of our proposed methods with baseline methods. Then, we make some interpretation assessments of the models.

4.1 Experimental setup

4.1.1 Datasets

To verify the effectiveness and efficiency of *TailRec*, we conducted experiments on three public datasets, for which we provide detailed statistics in Table 1. The first dataset, MovieLens[2)], is widely used in recommendation tasks. We converted the explicit ratings to implicit feedback and constructed each user's session by sorting according to their timestamp. To reduce the impact of noise data [29,30], we filtered out session lengths of 2 and items that appeared less than 5 times in both Diginetica and Yoochoose. For MovieLens, we performed basic pre-processing by filtering out items that appeared less than 20 times following [31]. However, the number of interaction frequencies followed a

2) See files.grouplens.org/datasets/movielens/ website.

Table 1 Statics of three public datasets after pre-processing

Datasets	Num. sequences	Num. items	Interactions	Gini index
MovieLens	876,162	23,514	25,417,546	0.87
YooChoose	1,286,641	34,867	9,859,412	0.84
Deginetica	130,601	44,237	699,208	0.54

heavy long-tailed distribution, with a large proportion of items showing low-frequency interactions. We set the maximum length of the former two datasets to 20 and 30 for MovieLens. We adopted the *leave-one-out* strategy to split these three datasets into training, validation, and testing, as described in [32].

4.1.2 Evaluation metrics

We employ two popular ranking metrics, including NDCG@K and Recall@K (N@K and R@K for short). Also, we evaluate the recommendation performance by ranking all candidate items without sampling. Besides, we also evaluate the diversity performance of compared methods by employing Tail@K (T@K for short), which measures how many tail items are in top-K ranked results. Note that Tail@K is defined by averaging all test cases, and is very prevalent in previous works [17]. Specifically, the computation of Tail@K can be described as

$$\mathrm{Tail@K} = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{|\mathrm{L}_{\mathrm{K}}^{\mathrm{Tail}}(u)|}{\mathrm{K}}, \tag{5}$$

where $\mathrm{L}_{\mathrm{K}}^{\mathrm{Tail}}(u)$ represents a subset of the the list of top-K ranked items for user u that belong to the set of long-tail items.

4.1.3 Compared methods

We conducted all experiments using the Transformer encoder in SASRec [19] as the backbone network. We refer to the variants of SASRec enhanced by our methods as *TailRec*. It is important to note that both SASRec and *TailRec* strictly follow the same hyper-parameter configuration to ensure meaningful comparisons. We employed two RNN-based approaches (*GRU4Rec* [21] and *NARM* [33]), one CNN-based SRS (*NextItNet* [34]), and two traditional matrix factorization and Markov chain approaches (*BPR-MF* [35] and *FPMC* [20]). Additionally, we compared our methods with *STAMP* [29], which also leverages a memory mechanism to represent user interests. In particular, we regarded *TailNet* [17], *MIRec* [36], and *CITIES* [18] as the competitive baselines in explicitly modeling long-tail distribution, proposing a preference mechanism to adjust the ranking results. To ensure a fair comparison, we replaced the sequence network with Transformer encoder [19] in *TailNet*.

4.1.4 Implement details

We trained all models on a single NVIDIA V100 GPU. To ensure a fair evaluation, we fine-tuned each hyper-parameter on the validation set. The embedding dimension was set to 64. For optimization, we used the Adam optimizer with a learning rate of 0.001 and a batch size of 256 for all datasets. All other model parameters either followed the authors' suggestions or were tuned on the validation sets. We trained each model until convergence and recorded the model parameters only when they achieved the highest ranking results on the validation datasets, and we report the results on test data. For SASRec, we set the layer number to 16 and the head number of multi-head attention to 4. We searched for the controllable parameters λ and γ from $0.1, 0.2, \ldots, 1.0$. We considered the half window size from $1, 2, 3, 4, 5$ and reported the best results. Following Pareto rule [17], we classified the entire item set into short-head items $\mathcal{I}^{head}$ and tail items $\mathcal{I}^{tail}$. Based on global interactions, we removed the bottom 10% interacted items from the item set $\mathcal{I}^{tail}$ to construct the item set $\mathcal{I}_e^{tail}$. More implementation details can be found in the publicly available codes[3].

4.2 Experimental results

4.2.1 Effectiveness measurement

In this section, we compare our proposed method with other baselines. Table 2 summarizes the performance of all models.

Among these baseline models, deep SRS could exceed traditional MC-based methods in most situations, reflecting the power of deep learning in model sequence dependence. Particularly, SASRec could yield more robust recommendation results than other approaches, indicating the powerful capacity of self-attentive architecture in sequential recommendation tasks. From these results, we also notice that the recommendation performance of backbone networks is greatly damaged by the preference adjustment strategies proposed in *TailNet*. To some extent, such experimental results keep consistent with claimed insight in [37], i.e., blindly removing the popularity bias would lose such an important signal, and further deteriorate model performance. Here, we keep alignment with such insights. For instance, some items exhibit higher popularity since they have intrinsic better properties. In contrast, *TailRec* can exhibit superior recommendation performance improvements than baselines in most cases. Such results reflect that: our proposed contextual representation component indeed can be useful for further improving embedding quality of tail items so as to further improve the capacity of benchmark recommenders in fitting user preference.

More than this, we can observe that *TailRec* also still improve the diversity of basis backbone networks. We hold the main reason behind such results is that the accurate embedding description is helpful to reflect the true user preferences in candidate items. Although *TailNet* can outperform the diversity performance of our methods, we hold such results are reasonable since the main intention of our works is not to eliminate the popularity bias.

[3] See github.com/Mingyue-Cheng/TailRec website.

Table 2 Performance comparison of all methods on sequential recommendation scenarios, where the best results are in bold and "*" denotes *TailRec* obtain gains over corresponding baselines

Methods	MovieLens			YooChoose			Deginetica		
	R@10	N@10	T@10	R@10	N@10	T@10	R@10	N@10	T@10
BPR-MF	0.0606	0.0284	0.0246	0.2404	0.1327	0.0831	0.0871	0.0421	**0.5978**
FPMC	0.1018	0.0532	0.0242	0.3224	0.1885	0.1170	0.2663	0.1600	0.1666
GRU4Rec	0.1521	0.0842	0.0520	0.4188	0.2540	0.1064	0.2119	0.1315	0.4153
NARM	0.1526	0.0844	0.0473	0.4178	0.2544	0.1034	0.2447	0.1473	0.4116
NextItNet	0.1538	0.0847	0.0429	0.4148	0.2496	0.1044	0.1475	0.0860	0.3922
STAMP	0.1261	0.0698	0.0492	0.4060	0.2467	0.1060	0.2563	0.1555	0.2808
SASRec	0.1641	0.0939	0.0507	0.4318	0.2634	0.1106	0.2777	0.1778	0.3231
TailNet	0.1505	0.0855	0.0795	0.4111	0.4367	**0.1311**	0.1989	0.1257	0.2366
MIRec	0.1655	0.0945	0.0533	0.4321	0.2633	0.1126	0.2800	0.1790	0.3477
CITIES	0.1624	0.0914	**0.0813**	0.4367	**0.2660**	0.1049	0.2929	0.1555	0.3333
TailRec	**0.1712***	**0.0977***	0.0666*	**0.4368***	0.2658*	0.1135*	**0.3170***	**0.1930***	0.3784*

4.2.2 Training convergence measurement

In this part, we aim to evaluate the efficiency of compared methods, which is specifically essential to large-scale recommendation scenarios. Specifically, we would report the training process of the *TailRec* and representative compared methods. For fair comparison, we omit the results of *MIRec* and *CITIES* since these two methods additionally adopt pre-training stage over recommenders. In addition, we only report the training curves in MovieLens while omitting the other two datasets for saving spaces. Since the testing loss aligns with the ranking results, we report the training process of ranking performance w.r.t. varying epochs in Fig. 4.

From these results, we can find that *TailRec*'s recommendation accuracy is constantly higher than its corresponding backbone methods. These results reflect that *TailRec* could accelerate its basic sequential recommender model through the entire process. Such results also evidence that improving the quality of tail item representation can significantly improve the efficiency of sequential recommenders. Also, we could find that the gap between the accuracy of the backbone model and *TailRec* keeps increasing during training. We hold it reasonable because the quality of the contextual embedding component would be gradually improved from the beginning stage to the last stage of the whole training process. With a small additional memory and computation cost, we also find that *TailRec* can save nearly less than 40% training time when reaching the same performance, indicating the potential accelerating ability of *TailRec*. We believe this is valuable because the expensive computational costs can be further saved, especially in large-scale recommendation scenarios.

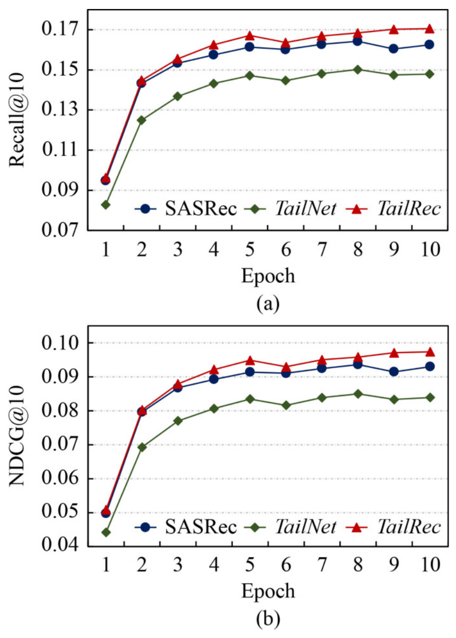

Fig. 4 (a) Training curves of TailRec and its baseline in SRS, evaluated on MovieLens data using the metric of Recall@10; (b) training curves of TailRec and its baseline in SRS, evaluated on MovieLens data using the metric of NDCG@10

4.2.3 Comparison w.r.t. evaluation scopes

To evaluate the effectiveness of our proposed model, *TailRec*, with respect to including tail items in historical behaviors or target items, we conducted extensive experiments by setting different evaluation scopes (as outlined in Section 2). We report the recommendation accuracy of our models and compare them with other methods in Table 3, using Recall@10 and NDCG@10 as evaluation metrics. For the results of evaluation scope w.r.t. "tail target items" and "w/ tail items", we surprisingly find that traditional low-order SRS, i.e., *FPMC*, could yield satisfying results in a way. These results reflect that traditional methods could be trained well with sparse sequential behaviors. By contrast, some deep sequential recommenders (like *GRU4Rec*, *NARM*, and *NextItNet*) might not perform well in data-poor scenarios since these methods usually require a considerable amount of training data. We also observe that our methods could consistently perform better than baselines in most cases, and achieve more improvements in these sparse data situations compared with evaluating all sequential behaviors. This is mainly because a larger proportion of tail items can be found in these instances, where the contextual representation module would be always called for boosting the benchmark recommenders.

Surprisingly, we also find that *TailRec* can even improve the performance of deep SRS in terms of these sequential behaviors without involving tail items. Such results are counterfactual in a way as those sequences contain no tail items, in which the contextual representation component

Table 3 Comparison of sequential recommendation performance with respect to different evaluation scopes, where the upper and below tables are Recall@10, NDCG@10, respectively. The best results are noted in bold while "*" denotes *TailRec* can obtain gains on baseline networks

Methods	Tail target items			w/ tail items			w/o tail items		
	ML	Yoo	Deg	ML	Yoo	Deg	ML	Yoo	Deg
BPR-MF	0.0095	0.1457	0.0157	0.0417	0.1876	0.0598	0.0806	0.2627	0.1437
FPMC	0.0077	**0.2531**	0.1112	0.0743	0.2831	0.2259	0.1308	0.3390	0.3518
GRU4Rec	0.0149	0.2048	0.0715	0.0974	0.3099	0.1546	0.2100	0.4648	0.3320
NARM	0.0134	0.1871	0.0872	0.0974	0.2962	0.1869	0.2109	0.4692	0.3654
NextItNet	0.0081	0.1835	0.0303	0.0992	0.2995	0.0929	0.2115	0.4635	0.2615
STAMP	0.0119	0.1963	0.0761	0.0763	0.2898	0.1966	0.1787	0.4551	0.3813
SASRec	0.0159	0.2274	0.1115	0.1078	0.3344	0.2195	0.2235	0.4730	0.3992
TailNet	0.0123	0.1644	0.0611	0.0948	0.2855	0.1392	0.2094	0.4642	0.3234
MIRec	0.0162	0.2248	0.1148	0.1084	0.3300	0.2236	0.2257	0.4753	0.3980
CITIES	0.0115	0.2229	0.1160	0.1083	0.3401	0.2380	0.2198	**0.4777**	**0.4091**
TailRec	**0.0202***	0.2382*	**0.1574***	**0.1136***	**0.3454***	**0.2749***	**0.2321***	0.4754*	0.4050
BPR-MF	0.0041	0.0726	0.0066	0.0198	0.1024	0.0285	0.0376	0.1455	0.0703
FPMC	0.0045	**0.1371**	0.0648	0.0388	0.1609	0.1357	0.0684	0.2001	0.2107
GRU4Rec	0.0085	0.1182	0.0454	0.0528	0.1822	0.0978	0.1174	0.2844	0.2020
NARM	0.0075	0.1075	0.0521	0.0526	0.1738	0.1129	0.1181	0.2885	0.2191
NextItNet	0.0044	0.1026	0.0174	0.0533	0.1737	0.054	0.1178	0.2818	0.1529
STAMP	0.0066	0.1109	0.0442	0.0413	0.1681	0.1192	0.0999	0.2799	0.2315
SASRec	0.0094	**0.1304**	0.0776	0.0608	0.1975	0.1456	0.1288	0.2913	0.2448
TailNet	0.0077	0.0989	0.0412	0.0527	0.1681	0.0907	0.1203	0.2861	0.1987
MIRec	0.1000	0.1287	0.0801	0.0611	0.1945	0.1482	0.1297	0.2924	0.2435
CITIES	0.0063	0.1208	0.0696	0.0594	0.1997	0.1594	0.1256	**0.2947**	**0.2504**
TailRec	**0.0116***	0.1298	**0.0938***	**0.0639***	**0.2012***	**0.1675***	**0.1335***	0.2931*	0.2463*

ML, Yoo and Deg represent MovieLens, Yoochoose and Deginetica respectively.

would be not called. This is probably because our contextual representation component can further improve the whole model parameters, including Transformer layers, indicating the whole SASRec model might be improved by our *TailRec* framework. To sum up, these observation implies that *TailRec* is able to make better use of the historical interactions with our enhanced representation mechanism, which thus alleviates the sparsity long-tail distribution problem for sequential recommenders.

4.2.4 Ablation studies

To gain a deeper understanding of how our *TailRec* works, we conducted an ablation study on three datasets. This study aimed to analyze the impact of the updating mechanism for the contextual representation component. When considering the updating mechanism for this component, two intuitive baselines come to mind: 1) updating the enhanced representation through back-propagation; and 2) freezing the contextual representation component without updating (shortened as "frozen"). We present the experimental results in Fig. 5. Our findings demonstrate that the weighted average approach achieves optimal performance compared to the other two strategies. We believe that this is probably because the weighted average method enables a smoother and more stable updating of the enhanced representation [28], which is beneficial for fusing the extracted contextual embedding information.

4.2.5 Applied in more recommenders

Our proposed enhanced representation can be easily integrated with other deep SRS to improve their performance. To verify this, we specified the proposed TailRec framework along with two baseline models, *GRU4Rec* and *NextItNet*, and compared the performance of these models with and without our contextual representation module. As shown in Fig. 6, we found that *GRU4Rec* and *NextItNet* achieved significant improvements after incorporating our enhanced representation, indicating the portability of this component. In the

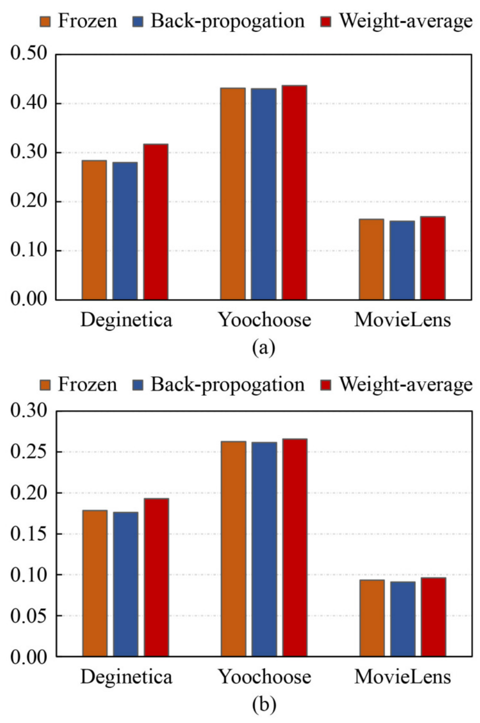

Fig. 5 The impacts of updating manner for contextual embedding module. (a) Recall@10; (b) NDCG@10

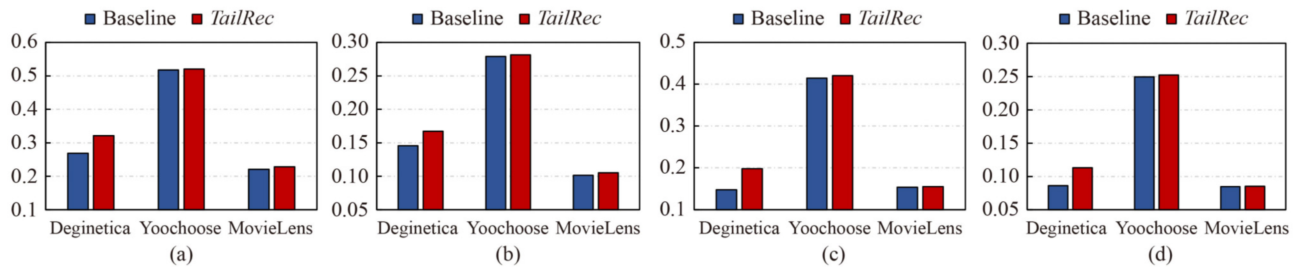

Fig. 6 The performance of GRU4Rec and NextItNet, both with and without enhanced representation on three datasets. (a) Testing Recall@10 on *GRU4Rec*; (b) Testing NDCG@10 on *GRU4Rec*; (c) Testing Recall@10 on *NextItNet*; (d) Testing NDCG@10 on *NextItNet*

future, we believe that a series of sequential recommenders can be enhanced to overcome obstacles related to poorly optimized representation of tail items.

4.2.6 Hyper-parameter sensitivity studies

In the following section, we explore the impact of hyper-parameters on final recommendation performance. Specifically, we examine the effect of varying the trade-off parameters λ and γ across different values: 0.1, 0.3, 0.5, 0.7, 0.9 for $t1, t2, t3, t4, t5$; the half window size s with values of 1, 2, 3, 4, 5 for $t1, t2, t3, t4, t5$; and the lower and upper bounds of the enhanced tail set as [50, 100], [50, 200], [50, 300], [50, 600], *all* for $t1, t2, t3, t4, t5$, where "all" denotes all items enhanced by our contextual representation module. For the sake of clarity and space-saving purposes, we only present the results of *TailRec* on MovieLens, as illustrated in Fig. 7.

The experimental outcomes reveal that *TailRec* is somewhat sensitive to the trade-off λ and the window size s, but highly sensitive to the trade-off γ and the scope of tail items. This is because these two hyper-parameters significantly affect the amount of tail items enhanced by our contextual representation module. Furthermore, our results demonstrate that the extra enhanced representation component plays a pivotal role in overall performance. In particular, we discovered that the model only generates sub-optimal outcomes when all items are enhanced. We speculate that this is due to the deviation of new representations of head items from the high-quality original representations as a result of the additional fusion of contextual representations.

5 Related work

In this section, we provide a concise overview of the relevant literature, focusing on two main aspects: 1) sequential recommenders and 2) long-tail recommendation.

5.1 Sequential recommendation

Understanding and characterizing user interests based on their historical behaviors is a central topic in recommender systems [2,38,39]. Recent research has devoted significant attention to sequential recommender systems (SRS) because users' historical interactions are sequentially dependent and naturally time-evolving. Early SRS works typically followed the Markov chain (MC) assumption to capture lower-order [20] or complex high-order sequence dependence [40]. With the breakthrough in deep learning [32], neural networks have become widely used in SRS. *GRU4Rec* [21], proposed as a pioneering work, modeled sequential recommendation by relying on the hidden representation of Gated Recurrent Units to record past behaviors. Subsequently, a series of variant methods were proposed, such as personalized SRS with hierarchical structures [41], context-aware SRS [42], data augmentation-based SRS [43], and memory networks [44]. While effective, these RNN-based models heavily rely on the hidden states of the entire past, which cannot be computed in parallel. In contrast, convolution-based [31,34] and pure attention-based SRS [19,24] have been widely studied because these methods can take full advantage of modern parallel processing resources and perform much better by stacking many repeated residual blocks [45]. In addition, modeling

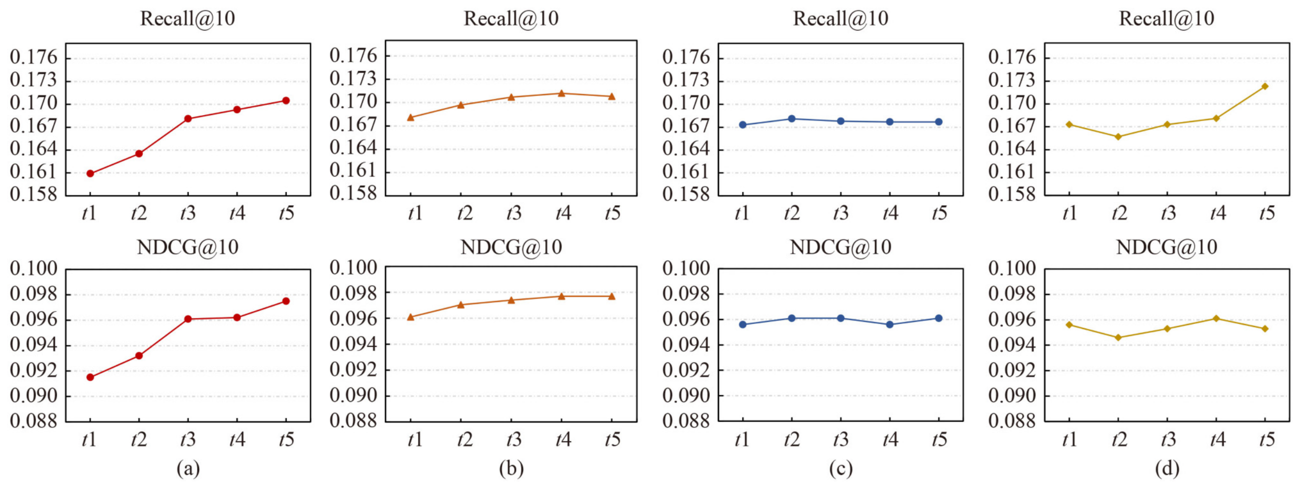

Fig. 7 Hyper-parameter sensitivity analysis of the proposed *TailRec* evaluated on SRS. (a) Trade-off γ; (b) trade-off α; (c) half window size s; (d) scope of enhanced item set

session-based recommendation with graph neural networks [2,46] recently becomes prevalent because it can capture both the global preference and the current interests of a session, simultaneously.

5.2 Long-tail recommendation

The long-tail distribution of items is a common characteristic of most recommendation datasets, as stated in [47]. In recent years, extensive efforts have been made in the field of RS to tackle the long-tail phenomenon in recommendation, as mentioned in [14]. One line of research aims to modify the loss function to guide the training of recommender models. For instance, weighting-based methods [48] have been proposed to re-weight the interactions in the training loss, with the weight set as the inverse of item popularity. In addition, previous works [49] attempt to solve the long-tail problem by performing ranking adjustments and proposing a regularization-based approach. Although these methods effectively alleviate the popularity issue [50], they often sacrifice recommendation accuracy by pushing the recommender towards the long-tail in a brute manner. Another line of research is to incorporate side information, such as user profiles and action types [51,52], to further address the problem. Aligning with the intention of these efforts above, several recent works [11,18] have been proposed to enhance the diversity of recommendation results.

6 Conclusion

In this work, we were motivated by the observation that poorly optimized tail item representations impede sequential recommendation results and training efficiency. To address this issue, we proposed a general framework named *TailRec* to boost sequential recommendation. In *TailRec*, we constructed an additional effective contextual representation module to focus on improving the quality of tail items. With this module, we can further leverage contextual information on cross-sequence behaviors to enhance the representation quality of tail items. We conducted extensive experiments to demonstrate the strengths of *TailRec*, which include: 1) greatly improved the effectiveness of SRS, 2) considerably accelerated the training of deep SR models, and 3) is intuitively simple, easy to implement, and applicable to a broad class of sequential recommender models. We hope that our findings and proposed framework can inspire more works in the future.

Acknowledgements This research was supported by grant from the National Key R&D Program of China (No. 2021YFF0901003). The authors also appreciate the partial support from Hefei AI Computing Center (Project Team).

Competing interests The authors declare that they have no competing interests or financial conflicts to disclose.

References

1. Wang S, Hu L, Wang Y, Cao L, Sheng Q Z, Orgun M A. Sequential recommender systems: challenges, progress and prospects. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. 2019, 6332−6338
2. Wu S, Tang Y, Zhu Y, Wang L, Xie X, Tan T. Session-based recommendation with graph neural networks. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence and 31st Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence. 2019, 43
3. Xu E, Yu Z, Li N, Cui H, Yao L, Guo B. Quantifying predictability of sequential recommendation via logical constraints. Frontiers of Computer Science, 2023, 17(5): 175612
4. Zhaok X, Liu H, Fan W, Liu H, Tang J, Wang C, Chen M, Zheng X, Liu X, Yang X. AutoEmb: automated embedding dimensionality search in streaming recommendations. In: Proceedings of 2021 IEEE International Conference on Data Mining. 2021, 896−905
5. Cheng M, Liu Q, Liu Z, Li Z, Luo Y, Chen E. FormerTime: hierarchical multi-scale representations for multivariate time series classification. In: Proceedings of the ACM Web Conference. 2023, 1437−1445
6. Cheng M, Liu Q, Liu Z, Zhang H, Zhang R, Chen E. TimeMAE: self-supervised representations of time series with decoupled masked autoencoders. 2023, arXiv preprint arXiv: 2303.00320
7. Sun Y, Yuan F, Yang M, Wei G, Zhao Z, Liu D. 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. 2020, 1299−1308
8. Chen L, Yuan F, Yang J, Ao X, Li C, Yang M. A user-adaptive layer selection framework for very deep sequential recommender models. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 3984−3991
9. Zhang S, Yao D, Zhao Z, Chua T S, Wu F. CauseRec: counterfactual user sequence synthesis for sequential recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021, 367−377
10. Yin J, Liu C, Wang W, Sun J, Hoi S C H. Learning transferrable parameters for long-tailed sequential user behavior modeling. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020, 359−367
11. Kim Y, Kim K, Park C, Yu H. Sequential and diverse recommendation with long tail. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. 2019, 2740−2746
12. Fan Z, Liu Z, Zhang J, Xiong Y, Zheng L, Yu P S. Continuous-time sequential recommendation with temporal graph collaborative transformer. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021, 433−442
13. Zipf G K. Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology. Xue C F, trans. Shanghai: Shanghai People’s Publishing House, 2016
14. Yin H, Cui B, Li J, Yao J, Chen C. Challenging the long tail recommendation. Proceedings of the VLDB Endowment, 2012, 5(9): 896–907
15. Liu Z, Cheng M, Li Z, Liu Q, Chen E. One person, one model-learning compound router for sequential recommendation. In: Proceedings of IEEE International Conference on Data Mining. 2022, 289−298
16. Wei T, Feng F, Chen J, Wu Z, Yi J, He X. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021, 1791−1800
17. Liu S, Zheng Y. Long-tail session-based recommendation. In: Proceedings of the 14th ACM Conference on Recommender Systems. 2020, 509−514
18. Jang S, Lee H, Cho H, Chung S. CITIES: contextual inference of tail-item embeddings for sequential recommendation. In: Proceedings of the 20th IEEE International Conference on Data Mining. 2020, 202−211
19. Kang W C, McAuley J. Self-attentive sequential recommendation. In: Proceedings of 2018 IEEE International Conference on Data Mining. 2018, 197−206
20. Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized Markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World Wide Web. 2010, 811−820

21. Hidasi B, Karatzoglou A, Baltrunas L, Tikk D. Session-based recommendations with recurrent neural networks. In: Proceedings of the 4th International Conference on Learning Representations. 2016
22. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017, 6000−6010
23. Cheng M, Liu Z, Liu Q, Ge S, Chen E. Towards automatic discovering of deep hybrid network architecture for sequential recommendation. In: Proceedings of the ACM Web Conference 2022. 2022, 1923−1932
24. Sun F, Liu J, Wu J, Pei C, Lin X, Ou W, Jiang P. BERT4Rec: sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019, 1441−1450
25. Cheng M, Yuan F, Liu Q, Ge S, Li Z, Yu R, Lian D, Yuan S, Chen E. Learning recommender systems with implicit feedback via soft target enhancement. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021, 575−584
26. Xue F, He X, Wang X, Xu J, Liu K, Hong R. Deep item-based collaborative filtering for top-N recommendation. ACM Transactions on Information Systems, 2019, 37(3): 33
27. Cai Y, Cui Z, Wu S, Lei Z, Ma X. Represent items by items: An enhanced representation of the target item for recommendation. 2021, arXiv preprint arXiv: 2104.12483
28. He K, Fan H, Wu Y, Xie S, Girshick R. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, 9726−9735
29. Liu Q, Zeng Y, Mokhosi R, Zhang H. 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. 2018, 1831−1839
30. De Souza Pereira Moreira G, Rabhi S, Lee J M, Ak R, Oldridge E. Transformers4Rec: bridging the gap between NLP and sequential/session-based recommendation. In: Proceedings of the 15th ACM Conference on Recommender Systems. 2021, 143−153
31. Tang J, Wang K. Personalized top-n sequential recommendation via convolutional sequence embedding. In: Proceedings of the 11th ACM International Conference on Web Search and Data Mining. 2018, 565−573
32. Cheng M, Yuan F, Liu Q, Xin X, Chen E. Learning transferable user representations with sequential behaviors via contrastive pre-training. In: Proceedings of 2021 IEEE International Conference on Data Mining (ICDM). 2021, 51−60
33. Li J, Ren P, Chen Z, Ren Z, Lian T, Ma J. Neural attentive session-based recommendation. In: Proceedings of 2017 ACM on Conference on Information and Knowledge Management. 2017, 1419−1428
34. Yuan F, Karatzoglou A, Arapakis I, Jose J M, He X. A simple convolutional generative network for next item recommendation. In: Proceedings of the 12th ACM International Conference on Web Search and Data Mining. 2019, 582−590
35. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L. BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence. 2012, 452−461
36. Zhang Y, Cheng D Z, Yao T, Yi X, Hong L, Chi E H. A model of two tales: dual transfer learning framework for improved long-tail item recommendation. In: Proceedings of the Web Conference 2021. 2021, 2220−2231
37. Zhao Z, Chen J, Zhou S, He X, Cao X, Zhang F, Wu W. Popularity bias is not always evil: disentangling benign and harmful bias for recommendation. IEEE Transactions on Knowledge and Data Engineering, 2022, doi: 10.1109/TKDE.2022.3218994
38. He X, Chua T S. Neural factorization machines for sparse predictive analytics. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2017, 355−364
39. Pi Q, Zhou G, Zhang Y, Wang Z, Ren L, Fan Y, Zhu X, Gai K. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2020, 2685−2692
40. He R, Fang C, Wang Z, McAuley J. Vista: a visually, socially, and temporally-aware model for artistic recommendation. In: Proceedings of the 10th ACM Conference on Recommender Systems. 2016, 309–316
41. Ying H, Zhuang F, Zhang F, Liu Y, Xu G, Xie X, Xiong H, Wu J. Sequential recommender system based on hierarchical attention network. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. 2018, 3926–3932
42. Gu Y, Lei T, Barzilay R, Jaakkola T. Learning to refine text based recommendations. In: Proceedings of 2016 Conference on Empirical Methods in Natural Language Processing. 2016, 2103−2108
43. Tan Y K, Xu X, Liu Y. Improved recurrent neural networks for session-based recommendations. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. 2016, 17−22
44. Huang J, Zhao W X, Dou H, Wen J R, Chang E Y. Improving sequential recommendation with knowledge-enhanced memory networks. In: Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. 2018, 505−514
45. Wang J, Yuan F, Chen J, Wu Q, Yang M, Sun Y, Zhang G. StackRec: efficient training of very deep sequential recommender models by iterative stacking. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021, 357−366
46. Liang Y, Song Q, Zhao Z, Zhou H, Gong M. BA-GNN: Behavior-aware graph neural network for session-based recommendation. Frontiers of Computer Science, 2023, 17(6): 176613
47. Brynjolfsson E, Hu Y J, Smith M D. From niches to riches: anatomy of the long tail. Sloan Management Review, 2006, 47(4): 67–71
48. Liang D, Charlin L, Blei D M. Causal inference for recommendation. In: Proceedings of Causation: Foundation to Application, Workshop at UAI. 2016
49. Abdollahpouri H, Burke R, Mobasher B. Controlling popularity bias in learning-to-rank recommendation. In: Proceedings of the 11th ACM Conference on Recommender Systems. 2017, 42−46
50. Adomavicius G, Kwon Y. Improving aggregate recommendation diversity using ranking-based techniques. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(5): 896–911
51. Bai B, Fan Y, Tan W, Zhang J. DLTSR: a deep learning framework for recommendations of long-tail web services. IEEE Transactions on Services Computing, 2020, 13(1): 73–85
52. Li J, Lu K, Huang Z, Shen H T. Two birds one stone: on both cold-start and long-tail recommendation. In: Proceedings of the 25th ACM International Conference on Multimedia. 2017, 898−906

Mingyue Cheng is currently pursuing PhD degree with the Anhui Province Key Laboratory of Big Data Analysis and Application (BDAA), School of Data Science, University of Science and Technology of China (USTC), China. Before that, he received BArts degree from Hefei University of Technology (HFUT), China. His general research interests span machine learning methods and their applications, especially focusing on sequence data mining and prediction, such sequential behaviors analysis, time series analysis. He has published more than 10 papers in referred journals and conference proceedings, e.g., the TKDE, WWW, SIGIR, ICDM. He also serves on the program committees of conferences, including SIGKDD and SIGIR.

Qi Liu received the PhD degree from University of Science and Technology of China (USTC), China in 2013. He is currently a professor in the School of Computer Science and Technology at USTC. His general area of research is data mining and knowledge discovery. He has published prolifically in refereed journals and conference proceedings (e.g., TKDE, TOIS, KDD). He is an Associate Editor of IEEE TBD and Neurocomputing. He was the recipient of KDD’18 Best Student Paper Award and ICDM’11 Best Research Paper Award. He is a member of the Alibaba DAMO Academy Young Fellow. He was also the recipient of China Outstanding Youth Science Foundation in 2019.

Wenyu Zhang is currently working toward the PhD degree in the School of Computer Science and Technology at University of Science and Technologyof China (USTC), China. He received his bachelor degree in communication engineering from Anhui Agricultural University (AHAU), China in 2021. His main research interests include recommender systems, and computer vision. He has published papers in referred conference proceedings, such as ACM MM’2021 and BMVC’2021.

Zhiding Liu received his BE degree in computer science from University of Science and Technology of China (USTC), China in 2021. He is currently working toward the ME degree in the School of Computer Science and Technology at University of Science and Technology of China (USTC), China. His main research interests include data mining, recommender systems, and time series representation learning. He has published papers in referred conference proceedings, such as WWW’2022 and ICDM’2022.

Hongke Zhao received the PhD degree from the University of Science and Technology of China (USTC), China. He is an associate professor with the College of Management and Economics, Tianjin University, China. His research interest includes data mining, data-driven management, knowledge and behavior computing. He has published more than 70 papers in refereed journals and conference proceedings, such as INFORMS Journal on Computing, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Evolutionary Computation, ACM Transactions on Information Systems, ACM SIGKDD, ACM WSDM, ACM SIGIR, IJCAI, AAAI, and IEEE ICDM. He was the recipient of Distinguished Dissertation Award Nomination of CAAI (2019), the Best Student Paper Award of CCML-2019.

Enhong Chen received the PhD degree from the University of Science and Technology of China (USTC), China. He is a professor and vice dean of the School of Computer Science, USTC, China. His general area of research includes data mining and machine learning, social network analysis, and recommender systems. He has published more than 100 papers in refereed conferences and journals, including the IEEE Transactions on Knowledge and Data Engineering, the IEEE Transactions on Mobile Computing, the IEEE Transactions on Industrial Electronics, the ACM Transactions on Knowledge Discovery from Data, ACM SIGKDD, IEEE ICDM, and NIPS. He was on program committees of numerous conferences including SIGKDD, ICDM, and SDM. He received the Best Application Paper Award on KDD’2008, the Best Research Paper Award on ICDM’ 2011, and the Best of SDM’2015. His research is supported by the National Science Foundation for Distinguished Young Scholars of China.