IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 29 May 2023, accepted 10 June 2023, date of publication 14 June 2023, date of current version 8 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3286021

RESEARCH ARTICLE

Long and Short-Term Interest Contrastive Learning Under Filter-Enhanced Sequential Recommendation

YI LI, CHANGCHUN YANG, TONGGUANG NI, YI ZHANG, AND HAO LIU
School of Computer Science and Artificial Intelligence, Changzhou University, Changzhou, Jiangsu 213164, China

Corresponding author: Changchun Yang (ycc@cczu.edu.cn)

This work was supported by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant KYCX22_3063.

ABSTRACT Existing self-supervised sequential recommendations face the problem of noisy interactions and sparse sequence data, and train models based only on item prediction losses, so they usually fail to learn an appropriate sequential representation. In this paper, to address the above problem, we propose long and short-term interest contrastive learning under filter-enhanced sequential recommendation (FLSCSR). Specifically, a filtering algorithm is used on the user's interaction sequences to attenuate the noisy information in the sequence data. Two independent encoders are used to model the user's long-term and short-term interests separately on the filter-based enhanced interaction sequences. Then user-specific gating mechanisms are constructed to capture the long-term and short-term interests tailored to the user's personalized preferences, which are incorporated into the attention network to achieve better learning of interest representations in sequence recommendations. In addition, representation alignment learning goals are proposed to minimize the discrepancy between long-term and short-term interest representations in personalized global contexts and local sequence representations. Experiments were conducted on three public and industrial datasets, where the FLSCSR model could obtain superior performance compared to the benchmark model: AUC improves by 0.76%-2.02%, GAUC improves by 0.55%-1.01%, MRR improves by 1.19%-2.09%, and NDCG@2 improves by 1.07%-2.26%.

INDEX TERMS Sequential recommendation, filtering algorithm, self-supervised learning, contrastive learning.

I. INTRODUCTION

With the popularity and development of Internet applications, the scale of data and information has exploded. Recommendation systems have become one of the largest AI commercialization application scenarios, providing users with personalized content services and improving the efficiency of information distribution. Traditional recommendations are usually algorithms based on collaborative filtering and content filtering of user behavior. The core idea is to consider the user's past behavior, such as the user's historical interaction data, search history, purchase history, rating history, etc. Based on these data, the user's level of interest in items that have not yet been interacted with is predicted, and the items that are most likely to be of interest to the user are recommended to the user. Unlike traditional recommendations, sequential recommendations take into account not only the user's historical behaviors, but also the sequential and temporal information between them. It usually uses sequence modeling techniques to learn patterns in user behavior sequences and predict users' possible future behaviors and interests to provide more personalized recommendations to users. The main difference between these two types of recommendations is whether the sequential and temporal information of historical behaviors is considered. Traditional recommendations only consider the static relationship between users and items, while sequential recommendations consider the dynamic evolution of user behavior. In addition, traditional

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Hao Chen.

recommendations are usually more applicable to sparse user-item matrices, while sequential recommendations are usually more applicable to long-term, continuous sequences of user behaviors.

With the wide application of deep learning methods in recommendation systems and user feedback prediction tasks, sequential recommendation [1], [2] has also adapted to the wave of deep learning, and numerous deep learning-based sequential recommendation algorithms have emerged [2], [3], [4], [5], [6], [7]. Some recent works [8], [9], [10], [11] use two different models to learn users' long and short-term (LS-term) interests separately. For example, Yu et al. [11] proposed the SLi-Rec model, which for the first time uses two separate encoders to extract users' LS-term interests separately and combines these two aspects of preferences through an attentional framework. Zheng et al. [12] pointed out that the SLi-Rec algorithm is not sufficient to distinguish between learned LS-term interests. On the contrary, it increases model redundancy, resulting in poor accuracy. Therefore, they further proposed the CLSR model, which is the most advanced method at present to monitor the separation of LS-term interests of users through contrastive learning.

Although the above methods achieve better sequential recommendation performance in terms of user modeling, feature interaction and representation learning, there are still some shortcomings that can be improved. Firstly, sequential recommendation models are usually constructed based on implicit feedback sequences, and due to the uncertainty of user behavior, behavioral sequences may contain noisy information and irrelevant interactions, thus interfering with the next interaction prediction [13]. Secondly, each user may only be interested in some specific attributes of the item, so user interests should be user-specific. Thirdly, user behavior data is very sparse. Previous methods usually use only item prediction tasks to learn model parameters, thus it is often affected by data sparsity and cannot learn sequence representation well [14], [15].

Therefore, in order to address the three issues mentioned above, we propose long and short-term interest contrastive learning under filter-enhanced sequential recommendation (FLSCSR). Based on the discussion and investigation of comparative experimental results, the FLSCSR model outperforms state-of-the-art sequential recommendation methods in terms of performance. The main contributions of this work can be summarized as follows:

1) This work introduces a filtering mechanism to attenuate noise information and extract meaningful features, thereby mitigating the impact of noise interactions in user behavior sequences on the contrastive learning-based recommendation model. It also prevents the generation of misleading self-supervised signals.
2) This work designs a user-specific gating mechanism that takes into account both the static features and dynamic context of users to capture personalized long-term and short-term interests tailored to their individual preferences.
3) This work proposes a representation alignment learning objective to minimize the discrepancy between personalized global context representations of long-term and short-term interests and local sequential representations. This aims to reduce information loss and distortion, assisting the model in better understanding user interests and preferences.

II. RELATED WORK

A. SEQUENTIAL RECOMMENDATION

The core idea of sequential recommendation is to learn the changes of users' interests from the sequence of interaction between users and items, so as to obtain more accurate dynamic recommendation effect. Traditional sequential recommendations are usually based on Markov chains and matrix decomposition methods, which treat the user's historical behaviour as static information and cannot effectively capture the user's long-term interests, making it difficult to handle complex sequence patterns. With the continuous development of deep learning, the sequential recommendation algorithm integrated with deep neural network has made great achievements. At the initial stage, the recurrent neural network [16], [17], [18], [19], [20], [21] and convolutional neural network [22], [23] were mainly used. These sequential recommendation models have shown outstanding advantages in modeling user behavior sequences, capturing temporal relationships in sequences, and extracting local features. Recently, with the application and rapid development of graph neural network in image, natural language processing and other fields, researchers have begun to use graph neural network for interactive sequence modeling [24], [25]. Graph neural networks have the advantages of capturing contextual information, handling heterogeneous data and modeling global contextual information in sequential recommendation algorithms. It provides a powerful modeling framework for personalized recommendations of recommender systems. The recent success of Transformer [26] has also driven the development of Transformer-based sequential recommendation models [1], [2]. These models can model users' interests and behavioral patterns more accurately, handle datasets of different sizes and more complex tasks. Transformer-based sequential recommendation models perform well in various recommendation scenarios.

Accurately modeling and distinguishing users' long-term and short-term interests is crucial [11]. However, traditional sequential recommendation and advanced deep learning models often fail to accurately differentiate users' long-term and short-term interests due to their use of unified representations that cannot fully capture the variations in users' interests. To address this issue, researchers have proposed methods to differentiate long-term and short-term interests. For example, Zhao et al. [8] modeled long-term interests using matrix factorization models and employed recurrent neural networks to model short-term interests. Yu et al. [11]

developed a variant of LSTM for modeling short-term interests and utilized asymmetric singular value decomposition to model long-term interests. However, these methods do not impose supervision on the learned interest representations, thus failing to guarantee accurate disentanglement of long-term and short-term interests. Therefore, we adopt the main framework of the state-of-the-art CLSR model in long-term and short-term modeling [12]. Our model separately models users' long-term and short-term interests and achieves their decoupling.

B. SELF-SUPERVISED LEARNING

In the early stages, self-supervised learning was primarily applied in computer vision [27] and natural language processing [28] domains. Subsequently, some studies introduced self-supervised learning into sequential recommendation to enhance the modeling of user behavior sequences and improve the effectiveness of sequence recommendation. For example, Zhou et al. [5] designed various self-supervised learning tasks based on background information. They first pre-trained the model and then fine-tuned it according to the recommendation task, achieving better recommendation performance. Xie et al. [7] utilized a contrastive learning framework to derive self-supervised signals from the original user behavior sequences, aiming to extract more meaningful user behavior patterns and further effectively encode user representations. Additionally, they proposed three data augmentation methods to construct self-supervised signals. However, these methods overlooked the differences between long-term and short-term interests, which are crucial for accurate recommendations.

Addressing the diverse user preferences, some researchers have studied disentangled representation learning. Ma et al. [29] proposed a variational autoencoder-based method to learn multiple user preferences. Wang et al. [30] utilized knowledge graphs to learn distinct user intents and distinguish them from each other. However, due to the lack of labeled data for unsupervised disentanglement, most of these methods failed to impose specific semantic constraints on the learned multiple representations, making them less effective. In contrast to existing unsupervised methods, our model introduces self-supervised signals and performs self-supervised learning by extracting contrastive tasks between learned representations and interest proxies from the original interaction sequences. This approach effectively distinguishes and learns long-term and short-term interests, thereby improving the accuracy and personalization of recommendations.

C. FILTERING ALGORITHMS

In order to capture the evolving patterns of user historical behavior, researchers have developed various deep neural network-based sequential recommendation models. Typical solutions include those based on recurrent neural networks [16], [31] and Transformer [1], [2], [5], which effectively leverage the historical information and exhibit significant performance in this task. However, these methods tend to overfit to all past peaks [32], [33], making them susceptible to noise and resulting in limited predictive performance. Therefore, a key challenge for sequential recommendation models is to accurately preserve historical information while reducing the influence of noise, achieving accurate and robust predictions. Existing research mainly focuses on utilizing auxiliary data to uncover users' underlying preferences and reduce noise in recommendation systems [34], [35], [36]. Auxiliary data can include external user behaviors, auxiliary item features, and additional feedback. These data can provide additional information about user interests, preferences, and behavior patterns, thereby enhancing the performance and effectiveness of the recommendation system. However, these methods often rely too heavily on users' past behavior, overlooking their latent interests and new preferences. Moreover, acquiring and processing auxiliary data requires significant effort, which is not always feasible in recommendation systems.

Recent studies have shown the advantages of Fourier Transform in terms of representation and characteristics in the time-frequency domain. Deep learning models based on Fourier Transform can transform input data into the frequency domain by performing spectral conversion on time-series signals, enabling learning and analysis based on the properties and features of Fourier Transform. Inspired by the theory of digital signal processing, Fourier Transform-based deep learning methods have gained increasing interest in the field of machine learning. For example, Woo et al. [37] employed Fourier Transform in time-series modeling to capture rich periodic information, providing an efficient learning paradigm in time-series analysis. Xu et al. [38] proposed a learning-based frequency selection method to identify trivial frequency components while removing redundant information, representing an exploration of using frequency-domain information for image processing. Zhou et al. [39] considered time-series prediction from the perspective of sequence compression and applied Fourier analysis to retain the part of the time series representation that correlates with low-frequency Fourier components, thereby eliminating the influence of noise. The research results demonstrate that deep learning models based on Fourier Transform can effectively capture the periodicity and frequency components of time-series signals, providing a powerful means of representation and modeling. Inspired by these characteristics, our model applies Fourier Transform to the task of long-term and short-term interest modeling in sequential recommendation. By utilizing Fourier Transform-based filtering algorithms, it obtains a sparse representation of the historical interaction sequence and removes redundant information, thereby improving the accuracy and quality of recommendations.

III. METHOD

Behavioral sequence data contains users' interest preferences, and mining this information can improve the accuracy

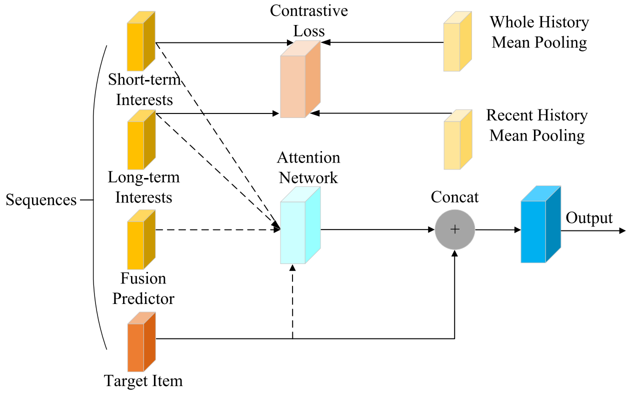

FIGURE 1. Network structure of CLSR.

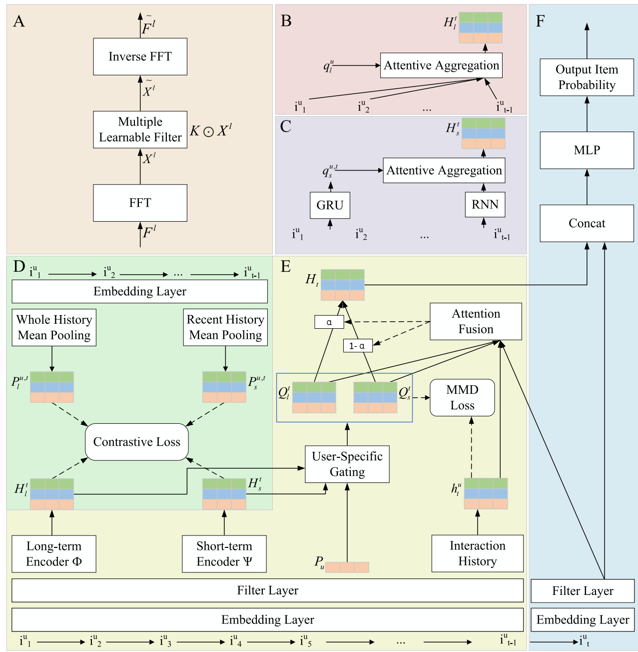

FIGURE 2. Network structure of FLSCSR.

of recommendation results. The sequences can be divided into long-term and short-term sequences based on the time span, which are used to characterize the long-term and short-term interests of users, respectively. Long-term interests reflect the interests and preferences of users that are sustained over time. It is usually more stable and less influenced by recent interactions. In other words, long-term interest can be inferred from the entire sequence of historical interactions. Therefore, the work uses user characteristics as input to the long-term interest representation, which contains the historical interactions of that user. Short-term interests, on the other hand, reflect the user's dynamic preferences that change rapidly based on recent interactions. Short-term interests evolve as users continue to interact with recommended products. For example, a user may build new interests after clicking on an item. At the same time, users may also gradually lose certain interests. That is, short-term interests are time-dependent variables. Therefore, in the evolution of short-term interest, the short-term interest of the current moment evolves from the short-term interest of the previous moment and is influenced by the user's interest representation of the previous moment and the previous target good. Most of the current work learns the unified representation of users' LS-term interests, or does not distinguish between the two. If an unsupervised approach is used to model users' LS-term interests, it is easy to mix the two aspects, resulting in worse performance. Therefore, the work chooses the CLSR architecture for the optimization of sequential recommendations, considering not only that it is the latest sequential recommendation algorithm available, but also that it obtains a decoupled representation of the user's long-term and short-term interests through a self-supervised approach.

A. CLSR

The overall structure of CLSR is shown in Fig. 1, which mainly consists of three parts: contrastive learning task, adaptive fusion and interaction prediction network. Specifically, firstly, two independent encoders are used to capture users' LS-term interests respectively. Then, based on the difference between users' LS-term interests, the average representation of the entire interaction history and the average representation of the last k interactions are used to construct pseudo-labels for long-term and short-term interests respectively, and decouple users' long-term and short-term interests by designing contrastive learning tasks. Finally, the user's LS-term interests are dynamically fused based on the target item and user interaction history using attention networks, and the final vector is spliced with the target item for prediction.

B. FLSCSR

The CLSR algorithm is not enough to capture the user's personalized preferences, and only uses the item prediction task learning model parameters, resulting in the inability to effectively represent the learning interest. In addition, since implicit feedback often does not represent users' true interests and sequential behaviors may contain noisy information, data denoising is also necessary to improve recommendation performance. In the work, we choose to use filters in the backbone network of CLSR to denoise the sequence behavior, and then introduce a user-specific gating mechanism to extract more comprehensive features of the user, and further design the loss function to better learn the sequence representation and avoid the impact of data sparsity problem. The improved algorithm is called FLSCSR in the work. Fig. 2 shows the overall schematic diagram.

Let the number of items be N and the number of users be M. $\{x_u\}_{u=1}^{M}$ denotes the set of interaction data for all users, i.e. the list of items. Each $x_u = \left[i_1^u, i_2^u, \ldots i_{Tu}^u\right]$ is the interaction data of user u sorted in chronological order, T_u represents the length of historical interactions of user u, and each commodity i_t^u is between $[1, N]$. The historical sequence of interactions $\{x_u\}_{u=1}^{M}$ for all users is used as input to the model and the output is the probability that a user will click on an item or not.

1) SEQUENCE FILTERING ENHANCEMENT

In order to reduce irrelevant interactions in the behavior sequence, the work obtains the sequence representation $E(x) \in R^{n \times d}$ after the user behavior sequence passes through the embedding layer, and then inputs it into the filter layer for data filtering enhancement. As shown in Fig. 2 (A), it will first undergo a fast Fourier transform transformation to the frequency domain, with the following equation.

$$X^l = \Re(F^l) \tag{1}$$

where, $F^l \in R^{n \times d}$, n is the length of the sequence, d is the dimension of the embedding, and l means that the sequence has passed through l learnable filter enhancement modules. If it is directly input from the embedded layer, then l=0, that is, $F^0 = E(x)$. $\Re(\bullet)$ denotes the Fast Fourier Transform and the transformed X^l is complex and $X^l \in C^{n \times d}$.

The sequence X^l after the Fourier transform is then modulated in the frequency domain by multiplying by a learnable filter $W \in C^{n \times d}$ to modulate the frequency spectrum. The formula is as follows,

$$\widetilde{X}^l = W \odot X^l \tag{2}$$

where $\odot$ is the element-wise multiplication. The filter K is called the learnable filter, which can be updated in each training round by random gradient descent to achieve adaptive filtering. The frequency domain filtered sequence is denoted by $\widetilde{X}^l$. At this time, the frequency component of the noise has been suppressed. Finally, the modulation spectrum $\widetilde{X}^l$ is transformed back into the time domain by the inverse fast Fourier transform and the sequence representation is updated. The formula is as follows,

$$E^f(x_j^u) = \widetilde{F}^l \leftarrow \Re^{-1}(\widetilde{X}^l) \in R^{n \times d} \tag{3}$$

where $\Re^{-1}(\bullet)$ represents the inverse fast Fourier transform, and $\widetilde{F}^l$ represents the user behavior sequence in the time domain after filtering. The fast Fourier transform and inverse fast Fourier transform operations can effectively reduce the noise from the recorded data and extract meaningful features from all frequencies.

2) SELF-SUPERVISED IMPLEMENTATION

The work first uses two independent encoders to learn separate representations of user interests at different time scales, i.e., long-term and short-term interests. In this case, long-term interests are learned from user feature embeddings and short-term interests are captured using recurrent neural networks. Specifically, query vectors are first generated for the encoders. The long-term interest encoder Φ uses user feature embeddings as query vectors. The short-term interest encoder Ψ takes the user's historical interaction sequences as input and uses gated recurrent units to capture the relationships over time. The final result is used as the query vector for the short-term interest encoder. The calculations are as follows,

$$\begin{aligned} q_l^u &= E(u), \\ q_s^{u,t} &= GRU(\{x_1^u, \ldots, x_t^u\}). \end{aligned} \tag{4}$$

Finally, the interaction sequence and the two query vectors q_l^u and $q_s^{u,t}$ obtained above are fed into the corresponding long-term encoder Φ and short-term encoder Ψ to obtain the long-term interest representations and short-term interest representations. The formulas are as follows,

$$\begin{aligned} H_l^t &= \Phi(q_l^u, \{x_1^u, \ldots, x_t^u\}), \\ H_s^t &= \Psi(q_s^{u,t}, \{x_1^u, \ldots, x_t^u\}). \end{aligned} \tag{5}$$

The specific process of the long-term encoder is shown in Fig. 2 (B). The work uses the attention mechanism to transform the sequence of user behaviors after multiplying the projection matrix with filtering enhancements into a new vector which is vectorially stitched with the long-term query vector. Then the weight of user's long-term interest is obtained by normalization of the multilayer perceptron. Finally, Weighted sum the filter-enhanced user behavior sequences to obtain the vector of long-term interest. The calculations are as follows,

$$\begin{aligned} v_j &= W_l E^f(x_j^u), \\ \alpha_j &= \tau_l(v_j \mid\mid q_l^u \mid\mid (v_j - q_l^u) \mid\mid (v_j \cdot q_l^u)), \\ a_j &= \frac{\exp(\alpha_j)}{\sum_{i=1}^t \exp(\alpha_i)}, \\ H_l^t &= \sum_{j=1}^t a_j E^f(x_j^u). \end{aligned} \tag{6}$$

where x_j^u is the interaction item of user u at moment j, $E^f(x_j^u)$ is the filter-enhanced user behavior sequence, W_l is the transformation matrix, τ_l denotes the multilayer perceptron network MLP, || denotes the splicing, a_j is the weight of the user's long-term interest obtained by normalization, and H_l^t is the long-term interest representation.

The process of short-term encoder is shown in Fig. 2 (C). First, short-term interest is captured on the filter-enhanced user behavior sequence using recurrent neural networks such as LSTM, GRU, Time-LSTM, etc. Since user behavior is sparse and easy to ignore past information, the work considers combining time intervals. Time-LSTM is used in the recurrent neural network model to add temporal information to reveal the relationship of user behavior. Similarly, the work uses the attention mechanism to multiply the projection matrix by the output sequence of the recurrent neural network, and then the new vector obtained. This vector and the short-term query vector are used to obtain the weights of the user's short-term interest. The weighted sum of the output sequences of the recurrent neural network is used to obtain the vector of short-term interest. The calculations are as follows,

$$\begin{aligned} \{o_1^u, \ldots, o_t^u\} &= \rho(\{E^f(x_1^u), \ldots, E^f(x_t^u)\}), \\ v_j &= W_s o_j^u, \\ H_s^t &= \sum_{j=1}^{t} b_j \cdot o_j^u. \end{aligned} \tag{7}$$

where ρ denotes the recurrent neural network model. o_j^u is the output sequence obtained from the recurrent neural network. b_j is the weight of the user's short-term interest obtained by normalization. H_s^t is the short-term interest representation.

The above user modeling process extracts the long-term and short-term interest representations of users in an unsupervised manner, which cannot ensure the separation of interests. Therefore, the work adds self-supervision to both long-term and short-term interest modeling. The agents of long-term and short-term interests are obtained from the interaction sequences to supervise the two interest encoders to achieve the decoupling of LS-term interests. Specifically, the work computes the average representation of the entire history of interactions as a proxy for long-term interest and uses the average representation of the last k interactions as a proxy for short-term interest. It is worth noting that the work computes the proxy only when the sequence length is greater than a threshold I_t, where the threshold I_t and the length k of the sequence of recent behaviors are hyperparameters in the method. If the whole sequence contains only a few items, there is no need to distinguish between long and short term. Then, a comparison task is used to monitor the similarity between interest representations and their corresponding interest agents to capture the LS-term interests of users more accurately. The calculations are as follows,

$$\begin{aligned} p_l^{u,t} &= \frac{1}{t}\sum_{j=1}^{t} E(x_j^u), \\ p_s^{u,t} &= \frac{1}{k}\sum_{j=1}^{k} E(x_{t-j+1}^u). \end{aligned} \tag{8}$$

where $p_l^{u,t}$ is the average representation over the entire interaction history, i.e. a proxy for long-term interest, and $p_s^{u,t}$ is the average representation using the last k interactions, i.e. a proxy for short-term interest.

Next, a contrastive learning task is constructed between the interest proxy and the interest output by the encoder, and the contrastive learning method is used to supervise that the representations under the same sequence are similar. The contrastive task is shown in Fig. 2 (D). Taking long-term interest as an example, the similarity between the long-term interest representation and the long-term interest proxy representation is greater than that with the short-term interest proxy representation. The four contrastive tasks are as follows,

$$\begin{aligned} &sim(H_l^t, p_l^{u,t}) > sim(H_l^t, p_s^{u,t}), \\ &sim(p_l^{u,t}, H_l^t) > sim(p_l^{u,t}, H_s^t), \\ &sim(H_s^t, p_s^{u,t}) > sim(H_s^t, p_l^{u,t}), \\ &sim(p_s^{u,t}, H_s^t) > sim(p_s^{u,t}, H_l^t). \end{aligned} \tag{9}$$

where $Sim()$ measures the similarity of embeddings. The model achieves stronger disentanglement through four contrasting tasks on the similarity between encoder output representations and interest agents. In this paper, we adopt either BPR or triplet loss to construct a loss function that captures the aforementioned constraint relationships, enabling contrastive learning. Formally, inner product or Euclidean distance is used to capture embedded similarity. Two loss functions are computed as follows,

$$\begin{aligned} Loss_{bpr}(a, p, q) &= \sigma(< a, q > - < a, p >), \\ Loss_{tri}(a, p, q) &= \max\{d(a, p) - d(a, q) + m, 0\}. \end{aligned} \tag{10}$$

where $\sigma(\bullet)$ is the activation function, $<, >$ is the inner product, d is the Euclidean distance and m is the boundary value. The four corresponding contrastive losses can be constructed by substituting H_l, H_s, p_l and p_s above with the following equations.

$$\begin{aligned} Loss_{con}^{u,t} = &f(H_l, p_l, p_s) + f(p_l, H_l, H_s) + f(H_s, p_s, p_l) \\ &+ f(p_s, H_s, H_l) \end{aligned} \tag{11}$$

where f can be either $Loss_{bpr}$ or $Loss_{tri}$.

3) USER-SPECIFIC GATING

As each individual user may only be interested in certain specific attributes of the project, long-term and short-term interests should be user-specific. The work captures long-term and short-term interests of personalized global context, tailored according to user preferences, through a user-specific gating mechanism. Taking the long-term representation of personalized global context as an example, the formula is as follows,

$$Q_l^t = H_l^t \otimes \sigma(H_l^t W_{g1} + P_u W_{g2}) \tag{12}$$

where $H_l^t \in R^{n\times d}$, $H_s^t \in R^{n\times d}$, $W_{g1} \in R^{d\times 1}$, $W_{g2} \in R^{d\times L}$, $\sigma(\bullet)$ is a sigmoid function and the user embedding P_u describes user characteristics. Similarly, the short-term representation of personalized global context for the user Q_s^t can be calculated through this formula. The next step is to design an auxiliary loss function for the model by characterizing the alignment target. The maximum mean discrepancy (MMD) is primarily used to measure the distance between two different but related feature distributions $x_i \in R^{m\times d}$ and $y_j \in R^{n\times d}$, i.e. the difference. The work therefore uses MMD to define the distance between the long-term and short-term interest representations of personalized global context (i.e. Q_l^t and Q_s^t)

and the representation of local sequence, defined as follows.

$$MMD[X, Y] = [\frac{1}{m^2}\sum_{i,j=1}^{m}\kappa(x_i, x_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n}\kappa(x_i, y_j) + \frac{1}{n^2}\sum_{i,j=1}^{n}\kappa(y_i, y_j)]^{\frac{1}{2}} \quad (13)$$

where $\kappa(\cdot,\cdot)$ is the kernel function, x_i and y_j denote the i-th row of X and the j-th row of Y, respectively. The Gaussian kernel with bandwidth ρ is used as the kernel function, i.e. $\kappa(x, y) = e^{-\frac{\|x-y\|^2}{2\rho^2}}$.

As shown in Fig. 2 (E), the historical sequence is processed through a GRU, resulting in a local sequence representation h_t^u. The model uses the maximum mean discrepancy to construct an auxiliary loss function for optimization. Specifically, it aims to minimize the distance between the long-term interest representation of personalized global context and the representation of local sequence, as well as minimize the distance between the short-term interest representation of personalized global context and the representation of local sequence. The formulas are as follows,

$$\begin{aligned} h_t^u &= GRU(\{E(x_1^u), \ldots, E(x_t^u)\}), \\ Loss_{MM}^{u,t} &= MMD(h_t^u, Q_l^t) + MMD(h_t^u, Q_s^t). \end{aligned} \quad (14)$$

4) DYNAMIC FUSION METHODS

Whether long-term or short-term interest plays a more important role in different user-item interactions depends on a variety of aspects, including the target item and the user's interaction history. For example, short-term interest is more important when users have been browsing similar items, while their behavior is largely driven by long-term interest when they move to a completely different item. Therefore, when user preferences change, the work uses an attention machine network to adapt user interests to predict dynamic changes in user preferences based on target goods and historical interactions.

As shown in Fig. 2 (F), the network adjusts weights to track the importance of long-term and short-term interests, accurately reflecting user preferences by integrating the fused user interests. The predicted items and user interest representations are then fed into a two-layer perceptron to obtain the prediction value. The calculations are as follows,

$$\begin{aligned} \alpha &= \sigma(\tau_f(h_t^u \,||\, E(x_{t+1}^u) \,||\, Q_l^t \,||\, Q_s^t)), \\ H^t &= \alpha \cdot Q_l^t + (1-\alpha) \cdot Q_s^t, \\ \hat{y}_{u,v}^{t+1} &= MLP(H^t \,||\, E(x_{t+1}^u)). \end{aligned} \quad (15)$$

where τ_f is a multilayer perceptron used for fusion. $\sigma(\bullet)$ is the sigmoid function. α represents the fusion weights for estimating based on local sequence representation, target item, and user's long-term and short-term personalized global contextual interests. H^t is the final representation of user interests.

TABLE 1. Statistics of the three datasets used in experiments.

Dataset	Users	Items	Instances	Average Length
Taobao	36,915	64,138	1,471,155	39.85
Amazon-Movies and TV	123,961	50,053	1,697,533	13.69
Yelp	30,431	20,033	316,354	10.40

5) MULTI TASK LEARNING

Multi-task modeling is widely used in advertising, recommendation and other scenario businesses in the current industrial field. Multi-task models can model common representations of goals and optimize goals for multiple tasks, thus improving predictions for multiple tasks. Since in real life industrial tasks are related to click-through rate (CTR) prediction, the work uses a negative log-likelihood loss function as the body of the loss function, with the following equation.

$$Loss_{rec}^{u,t} = -\frac{1}{N}\sum_{v\in O} y_{u,v}^{t+1}\log(\hat{y}_{u,v}^{t+1}) + (1-y_{u,v}^{t+1})\log(1-\hat{y}_{u,v}^{t+1}) \quad (16)$$

In addition to the main prediction task, the work incorporates two other auxiliary learning objectives. The final objective function of this model is calculated as follows.

$$Loss = \sum_{u=1}^{M}\sum_{t=1}^{T_u}(Loss_{rec}^{u,t} + \beta_1 Loss_{con}^{u,t} + \beta_2 Loss_{MM}^{u,t}) + \lambda\|\Theta\|_2 \quad (17)$$

where β_1 and β_2 are hyperparameters. $\lambda\|\Theta\|_2$ denotes L2 regularization to solve the overfitting problem.

IV. EXPERIMENTS AND DISCUSSION

A. EXPERIMENTAL PLATFORM

The experiment built a network using the TensorFlow1.15.2 deep-learning framework with the Ubuntu 16.04 operating system and Pycharm. The hardware configuration is as follows: dual parallel Nvidia Geforce GTX 1080 processor, 8G video memory, and 16G memory.

B. HYPERPARAMETER SETTING

The models used in the work are optimized by Adam. The initial learning rate is set to 0.001, and Xavier initialization is used to initialize parameters. The datasets are all fed at batch size 400, and the embedding size is fixed at 40 for all models.

C. DATASETS

The work evaluates the recommendation performance of public e-commerce data, and Table 1 summarizes the basic statistics for the three datasets.

Taobao: The dataset was collected from the largest e-commerce platform in China and is widely used in recommendation studies [40], [41]. Click-through data from 25 November to 3 December 2017 was retained, and users with less than 10 interactions were screened. Based on existing work, the first 7 days are used as the training set, the

8th day as the validation set, and the last day as the test set in the work.

Amazon: The dataset contains many Amazon product reviews and metadata, and it is widely used as a benchmark dataset in the field of recommendation systems. The work conducts experiments on a subset named Movies and TV. The dataset is rich in user behaviors, and contains comments that can reveal customers' shopping behaviors.

Yelp: The dataset is used for personal, educational and academic purposes. Since it is very large, the less active users and merchants are filtered by a 10-core setting, that is, ensure that each user and product in the dataset has at least 10 reviews.

D. EVALUATION METRICS

To evaluate the performance of each model, four metrics were selected as model evaluation metrics in this work. These include two widely used accuracy metrics, AUC and GAUC, and two widely used ranking metrics, MRR and NDCG, which are defined as follows.

AUC: AUC indicates the proportion of correctly classified samples to the total number of samples and reflects the ability of the classification model to rank the samples.

GAUC: GAUC is an improved version of AUC, which is a weighted average of each user's AUC, where the weight is the number of clicks he has made. It analyzes from different user personalized recommendations.

MRR: It is the ranking of the target item in the actual recommendation list, focusing on whether the recommended item is at the top.

NDCG@K: Considering the actual relevance and ranking order of each item, it is used to evaluate the quality of the ranking results. In the work, k is set to 2, which is a widely used setting in existing work.

E. BASELINE METHODS

The work compares the proposed FLSCSR model with the following baseline methods.

NCF [42]: The approach uses deep learning to model user and project characteristics, giving the model non-linear representational capabilities. Specifically, a multi-layer perceptron is used to learn the user-item interaction function and thus derive the resultant values.

DIN [43]: A local activation unit is used to increase the weight of historical information related to candidate advertisements, and attention mechanism is introduced to fully mine the information in the user's historical behavior sequence.

LightGCN [44]: The method proposes a lightweight graph convolutional network for collaborative filtering, which greatly simplifies the model design.

Caser [22]: The method uses vertical and horizontal convolution to capture the sequential behaviour patterns of the user for recommendations.

GRU4Rec [16]: The method uses gated recurrent units to capture sequential correlations and make recommendations.

DIEN [45]: The method uses recurrent neural networks to extract user interests more accurately. And further uses recurrent neural networks and attention mechanism to capture the dynamic changes of interest to encode the sequence of items.

CLSR [12]: The method uses a contrastive learning framework to separate users' long-term and short-term interests.

F. PERFORMANCE COMPARISON

1) EXPERIMENTAL COMPARISON OF DIFFERENT METHODS

In order to verify the recommendation performance of the proposed model, experiments were conducted on three datasets, Taobao, Amazon-Movie and TV and yelp. Table 2 shows the performance comparison of the different methods.

Table 2 shows the recommendation performance of eight different methods on three datasets, and the results show that the algorithm in this study has a more outstanding performance compared to other algorithms. Among them, the AUC metric of the FLSCSR model reaches 89.17% on the Taobao dataset, which is 1.16 to 17.97 percentage points higher than other algorithms. On the Amazon-Movie and TV dataset, the FLSCSR model achieves an AUC metric of 86.22%, which is 2.02 to 10.58 percentage points higher than other algorithms. On the yelp dataset, the FLSCSR model achieves an AUC metric of 93.69%, which is 0.76 to 11.12 percentage points higher than other algorithms. In summary, the overall performance of the improved model is greatly improved compared to other algorithms.

2) ABLATION STUDY

The improved FLSCSR algorithm is based on the original backbone network with the addition of a filter layer (denoted as Filter), a user-specific gating mechanism (denoted as Gating) and an auxiliary loss (denoted as MMD). In order to be able to analyze more intuitively the improvement of the improved FLSCSR compared to the original algorithm, the work conducts ablation studies on two datasets, Taobao and Amazon-Movie and TV, and the specific results are shown in Table 3. As shown in Table 3, Experiment 1 shows the experimental results of the CLSR algorithm without the addition of Filter, Gating and MMD. On the Taobao dataset, the AUC metrics and MRR metrics of the model are 88.01% and 42.74%, respectively. On the Amazon-Movie and TV dataset, they are 84.2% and 32.08%, respectively.

One key challenge faced by sequential recommendation models is how to accurately preserve historical information while reducing the interference of noise, in order to achieve accurate and robust predictions. Recent research has shown that deep learning models based on Fourier Transform can effectively capture the periodicity and frequency components of time series signals. Inspired by this characteristic, our model applies Fourier Transform to the task of modeling long-term and short-term interests in sequential

TABLE 2. Comparison of the performance of the different methods.

Dataset	Taobao				Amazon-Movie and TV				yelp			
Method	AUC	GAUC	MRR	NDCG@2	AUC	GAUC	MRR	NDCG@2	AUC	GAUC	MRR	NDCG@2
NCF	0.712	0.7198	0.1405	0.0785	0.7564	0.765	0.2072	0.144	0.8315	0.8418	0.7876	0.7644
DIN	0.7567	0.8549	0.319	0.2459	0.7845	0.8452	0.3368	0.2734	0.8454	0.9284	0.8988	0.9017
LightGCN	0.7513	0.7523	0.1684	0.102	0.7815	0.7891	0.2348	0.1683	0.8257	0.8164	0.7664	0.7291
Caser	0.8384	0.8559	0.3603	0.2982	0.7976	0.8291	0.2994	0.2336	0.9124	0.9259	0.8815	0.8849
GRU4REC	0.8601	0.866	0.3991	0.3422	0.8242	0.843	0.3325	0.2692	0.9195	0.9259	0.8832	0.8874
DIEN	0.831	0.8713	0.3978	0.3373	0.8166	0.843	0.3273	0.2643	0.8921	0.9202	0.8907	0.8899
CLSR	0.8801	0.8864	0.4274	0.3689	0.842	0.8486	0.3208	0.2533	0.9293	0.9283	0.8834	0.889
FLSCSR	**0.8917**	**0.8919**	**0.4402**	**0.3827**	**0.8622**	**0.8587**	**0.3417**	**0.2759**	**0.9369**	**0.9365**	**0.8953**	**0.8997**

TABLE 3. Results of ablation experiments at both AUC and MRR metrics.

Experiment	Filter	Gating	MMD	Taobao		Amazon-Movie and TV	
				AUC	MRR	AUC	MRR
1	-	-	-	0.8801	0.4274	0.842	0.3208
2	✓	-	-	0.8868	0.4301	0.8454	0.3278
3	-	✓	-	0.8875	0.4321	0.8552	0.3342
4	-	✓	✓	0.8912	0.4348	0.8608	0.3363
5	✓	✓	✓	0.8917	0.4402	0.8622	0.3417

recommendation. In Experiment 2, we added a Filter module to the CLSR algorithm, which utilizes a Fourier Transform-based filtering algorithm. On the Taobao dataset, the AUC metrics and MRR metrics of the improved model improved by 0.67% and 0.27%, respectively, over those of the original model. On the Amazon-Movie and TV dataset, the improvement was 0.34% and 0.7%, respectively. The experimental results demonstrate that the method achieves more accurate representation and prediction capabilities in sequential recommendation tasks by utilizing the Fourier Transform-based filtering algorithm. This indicates that the Fourier Transform-based filtering algorithm provides a powerful means of representation and modeling in sequential recommendation, and it has the potential to enhance recommendation performance.

This model considers the entire sequence information and models it, capturing more global sequence representations and contextual information. As the global context information is expected to be user-specific, supplementing the static features of the user profile is crucial for personalized recommendation systems. Therefore, this model incorporates a user-specific gating mechanism. Experiment 3 is the addition of the Gating module to the CLSR algorithm. The model also shows better recommendation performance after adding the user-specific gating mechanism. On the Taobao dataset, the AUC and MRR metrics of the improved model improved by 0.74% and 0.47%, respectively, over those of the original model. On the Amazon-Movie and TV dataset, the improvement is 1.32% and 1.34%, respectively. This demonstrates that integrating both the static features of the users and dynamic context can enhance the accuracy of the recommendation system and improve its performance and user experience.

In recent studies, the GCL4SR model has employed maximum mean discrepancy to assist in capturing the global context in sequential representations, enabling a better understanding of user interests and preferences, and generating personalized recommendations that align more closely with user preferences. Inspired by auxiliary learning objectives, this model introduces a representation alignment objective that integrates both global and local information. This module aims to reduce information loss and distortion, thereby enhancing the accuracy, personalization, and information consistency of the recommendation system. Experiment 4 is a further addition of the MMD loss module to the model of Experiment 3. The improved model shows outstanding sequence recommendation performance on both datasets. On the Taobao dataset, the AUC metrics and MRR metrics improved by 1.11% and 0.74%, respectively, over those of the original model. On the Amazon-Movie and TV dataset, the improvement is 1.88% and 1.55%, respectively. The experimental results demonstrate that minimizing the distance between personalized global contextual long-term and short-term interest representations and local sequential representations using MMD contributes to improved accuracy of the recommendation system, resulting in recommendations that better align with the users' actual needs.

Experiment 5 shows the experimental results of the final improved model under the conditions of adding Filter, Gating and MMD. On the Taobao dataset, the AUC metrics and MRR metrics of the FLSCSR model improve by 1.16% and 1.28%, respectively, over those of the original model. On the Amazon-Movie and TV dataset, the AUC and MRR metrics of the FLSCSR model improve by 2.02% and 2.09%, respectively, over those of the original model. It can be seen that the improved FLSCSR model has better performance in recommendation performance after data denoising, adding user-specific gating mechanism and auxiliary loss to the original backbone network.

V. CONCLUSION AND FUTURE WORK

Noisy interactions and data sparsity are pressing problems in the field of sequence recommendation, and most of the current research methods use only item-based prediction loss, which cannot effectively obtain sequence representations. Therefore, we propose long and short-term interest contrastive learning under filter-enhanced sequential recommendation with targeted improvements on the basis of the CLSR model. The work uses a filtering mechanism to attenuate the influence of neural networks by noise, constructs a user-specific gating mechanism to enable user interests to reflect unique user preferences. Furthermore, a representation alignment learning objective is proposed to better learn sequential representation. Extensive experiments on public datasets have shown that the FLSCSR model has better recommendation performance.

Although the method proposed in the work has achieved good results, there are still some areas for improvement. Future work will focus on designing alternative ways of extracting proxies of interest and exploring new auxiliary function learning tasks to further enhance the performance of FLSCSR. In addition, the application of the proposed method to the industrial field is another important work in the future.

CONFLICTS OF 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.

ACKNOWLEDGMENT

The authors would like to thank the reviewer's comments which helped to improve this article.

REFERENCES

[1] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Singapore, Nov. 2018, pp. 197–206.

[2] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Beijing, China, Nov. 2019, pp. 1441–1450.

[3] J. Li, Y. Wang, and J. McAuley, "Time interval aware self-attention for sequential recommendation," in *Proc. 13th Int. Conf. Web Search Data Mining*, Houston, TX, USA, Jan. 2020, pp. 322–330.

[4] J. Ma, C. Zhou, H. Yang, P. Cui, X. Wang, and W. Zhu, "Disentangled self-supervision in sequential recommenders," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery, Data Mining*, Aug. 2020, pp. 483–491.

[5] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J.-R. Wen, "S3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization," in *Proc. 29th ACM Int. Conf. Inf., Knowl. Manage.*, Oct. 2020, pp. 1893–1902.

[6] Z. Fan, Z. Liu, J. Zhang, Y. Xiong, L. Zheng, and P. S. Yu, "Continuous-time sequential recommendation with temporal graph collaborative transformer," in *Proc. 30th ACM Int. Conf. Inf., Knowl. Manage.*, Oct. 2021, pp. 433–442.

[7] X. Xie, F. Sun, Z. Liu, S. Wu, J. Gao, J. Zhang, B. Ding, and B. Cui, "Contrastive learning for sequential recommendation," in *Proc. IEEE 38th Int. Conf. Data Eng. (ICDE)*, May 2022, pp. 1259–1273.

[8] W. Zhao, B. Wang, J. Ye, Y. Gao, M. Yang, and X. Chen, "PLASTIC: Prioritize long and short-term information in top-*n* recommendation using adversarial training," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Stockholm, Sweden, Jul. 2018, pp. 3676–3682.

[9] M. An, F. Wu, C. Wu, K. Zhang, Z. Liu, and X. Xie, "Neural news recommendation with long and short-term user representations," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Florence, Italy, 2019, pp. 336–345.

[10] F. Lv, T. Jin, C. Yu, F. Sun, Q. Lin, K. Yang, and W. Ng, "SDM: Sequential deep matching model for online large-scale recommender system," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Beijing, China, Nov. 2019, pp. 2635–2643.

[11] Z. Yu, J. Lian, A. Mahmoody, G. Liu, and X. Xie, "Adaptive user modeling with long and short-term preferences for personalized recommendation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Macao, China, Aug. 2019, pp. 4213–4219.

[12] Y. Zheng, C. Gao, J. Chang, Y. Niu, Y. Song, D. Jin, and Y. Li, "Disentangling long and short-term interests for recommendation," in *Proc. ACM Web Conf.*, Apr. 2022, pp. 2256–2267.

[13] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Singapore, Nov. 2017, pp. 1419–1428.

[14] J. K. Arthur, C. Zhou, J. Osei-Kwakye, E. A. Mantey, and Y. Chen, "A heterogeneous couplings and persuasive user/item information model for next basket recommendation," *Eng. Appl. Artif. Intell.*, vol. 114, Sep. 2022, Art. no. 105132.

[15] J. K. Arthur, C. Zhou, E. A. Mantey, J. Osei-Kwakye, and Y. Chen, "A discriminative-based geometric deep learning model for cross domain recommender systems," *Appl. Sci.*, vol. 12, no. 10, p. 5202, May 2022.

[16] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," 2015, *arXiv:1511.06939*.

[17] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *Proc. 11th ACM Conf. RecSys*, Como, Italy, Aug. 2017, pp. 130–137.

[18] C. Ma, P. Kang, and X. Liu, "Hierarchical gating networks for sequential recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery, Data Mining*, Anchorage, AK, USA, Jul. 2019, pp. 825–833.

[19] A. Yan, S. Cheng, W.-C. Kang, M. Wan, and J. McAuley, "CosRec: 2D convolutional neural networks for sequential recommendation," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Beijing, China, Nov. 2019, pp. 2173–2176.

[20] L. Zheng, Z. Fan, C.-T. Lu, J. Zhang, and P. S. Yu, "Gated spectral units: Modeling co-evolving patterns for sequential recommendation," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Paris, France, Jul. 2019, pp. 1077–1080.

[21] B. Peng, Z. Ren, S. Parthasarathy, and X. Ning, "HAM: Hybrid associations models for sequential recommendation," 2020, *arXiv:2002.11890*.

[22] J. Tang and K. Wang, "Personalized top-N sequential recommendation via convolutional sequence embedding," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Marina Del Rey, CA, USA, Feb. 2018, pp. 565–573.

[23] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He, "A simple convolutional generative network for next item recommendation," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Melbourne, VIC, Australia, Jan. 2019, pp. 582–590.

[24] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, Honolulu, HI, USA, Jul. 2019, pp. 346–353.

[25] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, J. Fang, and X. Zhou, "Graph contextualized self-attention network for session-based recommendation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Macao, China, Aug. 2019, pp. 3940–3946.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Long Beach, CA, USA, Dec. 2017, pp. 1–11.

[27] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, Nov. 2021, doi: 10.1109/TPAMI.2020.2992393.

[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[29] J. Ma, C. Zhou, P. Cui, H. Yang, and W. Zhu, "Learning disentangled representations for recommendation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Vancouver, BC, Canada, 2019, pp. 1–12.

[30] X. Wang, T. Huang, D. Wang, Y. Yuan, Z. Liu, X. He, and T.-S. Chua, "Learning intents behind interactions with knowledge graph for recommendation," in *Proc. Web Conf.*, Apr. 2021, pp. 878–887.

[31] Q. Liu, S. Wu, D. Wang, Z. Li, and L. Wang, "Context-aware sequential recommendation," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Barcelona, Spain, Dec. 2016, pp. 1053–1058.

[32] R. Caruana, S. Lawrence, and C. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," in *Proc. NeurIPS*, vol. 13, 2000, pp. 381–387.

[33] J. Lever, M. Krzywinski, and N. Altman, "Points of significance: Model selection and overfitting," *Nature Methods*, vol. 13, no. 9, pp. 703–705, 2016.
[34] Y. Kim, A. Hassan, R. W. White, and I. Zitouni, "Modeling dwell time to predict click-level satisfaction," in *Proc. 7th ACM Int. Conf. Web Search Data Mining*, Feb. 2014, pp. 193–202.
[35] H. Lu, M. Zhang, W. Ma, C. Wang, F. Xia, Y. Liu, L. Lin, and S. Ma, "Effects of user negative experience in mobile news streaming," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Paris, France, Jul. 2019, pp. 705–714.
[36] B. Yang, S. Lee, S. Park, and S.-G. Lee, "Exploiting various implicit feedback for collaborative filtering," in *Proc. 21st Int. Conf. World Wide Web*, Apr. 2012, pp. 639–640.
[37] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting," 2022, *arXiv:2202.01575*.
[38] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, and F. Ren, "Learning in the frequency domain," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1737–1746.
[39] T. Zhou, Z. Ma, X. Wang, Q. Wen, L. Sun, T. Yao, W. Yin, and R. Jin, "FiLM: Frequency improved Legendre memory model for long-term time series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 12677–12690.
[40] H. Zhu, D. Chang, Z. Xu, P. Zhang, X. Li, J. He, H. Li, J. Xu, and K. Gai, "Joint optimization of tree-based index and deep model for recommender systems," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Vancouver, BC, Canada, Dec. 2019, pp. 1–10.
[41] Q. Pi, W. Bian, G. Zhou, X. Zhu, and K. Gai, "Practice on long sequential user behavior modeling for click-through rate prediction," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery, Data Mining*, Anchorage, AK, USA, Jul. 2019, pp. 2671–2679.
[42] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. WWW*, Perth, WA, Australia, Apr. 2017, pp. 173–182.
[43] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery, Data Mining*, London, U.K., Jul. 2018, pp. 1059–1068.
[44] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 639–648.
[45] G. Zhou, N. Mou, Y. Fan, Q. Pi, W. Bian, C. Zhou, X. Zhu, and K. Gai, "Deep interest evolution network for click-through rate prediction," in *Proc. AAAI Conf. Artif. Intell.*, Honolulu, HI, USA, 2019, pp. 5941–5948.

YI LI was born in June 1997. She is currently pursuing the master's degree with the School of Computer Science and Artificial Intelligence, Changzhou University. Her current research interest includes recommendation systems.

CHANGCHUN YANG received the master's degree from the East China University of Science and Technology. He is currently the Dean of the School of Microelectronics and Control Engineering, Changzhou University. He has published more than 50 academic papers. His research interests include database systems, data mining, and big data analytics.

TONGGUANG NI received the Ph.D. degree from Jiangnan University. He is currently an Associate Professor with the School of Computer Science and Artificial Intelligence, Changzhou University. His research interests include pattern recognition and artificial intelligence.

YI ZHANG was born in October 1998. He is currently pursuing the master's degree with the School of Computer Science and Artificial Intelligence, Changzhou University. His current research interest includes recommender systems.

HAO LIU was born in October 1999. He is currently pursuing the master's degree with the School of Computer Science and Artificial Intelligence, Changzhou University. His current research interest includes sentiment analysis.