Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/13/20230723

Multi-scale filter-enhanced transformer for sequential recommendation

Xiyu Cui

College of Letters & Science, University of Wisconsin–Madison, United States, 53706

xcui58@wisc.edu

Abstract. Recent research has shown that self-attention mechanisms improve Sequential Recommender Systems (SRS) by capturing sequential associations with the interactions. Nevertheless, existing work still needs to address two critical limitations. Firstly, the behavior of users in the original sequences contains various preference signals that are implicit and noisy and hard to reflect the user's intentions fully. As a result, it would deteriorate the representation of their true intentions to model all interactions. Secondly, most models only model single-scale interaction sequences and ignore the multi-scale feature relationships of the sequences. In order to address these limitations, the paper proposes MFTSRec (Multi-scale Filter Enhanced Transformer Sequential Recommender), which can weaken those interactions irrelevant to the users' intentions from their implicit feedback and adaptively focus on the user's multi-scale intentions. Besides, this paper also does extensive experiments on four benchmark datasets and further demonstrates the effectiveness and robustness of MFTSRec compared to the state-of-the-art model.

Keywords: sequential recommender system, multi-scale user interactions, transformer, fourier transform, multi-intentions modeling.

1. Introduction

Recommender systems facilitate user' information search by providing customers with personalized information and suggestions. They have played a key role in diverse online applications, such as E-commerce applications and music-video recommendations [1-4]. Most systems attempt to infer the user's current intentions and thus recommend products that may interest to the user based on the user's historical interaction data. Traditional recommender system focuses on collaborative filtering (CF), which recommends items of interest to users through the preferences of groups with similar interests and shared experiences. The standard paradigm for CF is to take users and items as embedded parameters and learn the embedded parameters of users and items by reconstructing historical user-item interactions. More recent neural recommendation models (e.g., NCF [5] and NGCF [6]) use the same embedding components with enhanced interactions as well as higher-order connectivity.

Different from traditional CF recommender systems, sequential recommender systems attempt to construct and analyze sequential patterns to capture features such as users' long and short-term preferences, goals, and consumption trends of items, thus enabling more accurate, diverse, and dynamic recommendations. Markov chains, one of the early works on sequential recommendations, embed transition information between adjacent interactions into the recommended item potential factors [7].

DOI: 10.54254/2755-2721/13/20230723

Later, with the development of deep learning, more and more network architectures are being applied to sequence recommendations, such as convolutional neural networks (CNN), recurrent neural networks (RNN). This paper proposes the Multi-scale Filtering Enhanced Transformer **S**equential Recommender (MFTSRec) framework. Firstly, to reduce noisy signals in user interaction sequences, MFTSRec designs a filter layer that filters certain frequency signals in the frequency domain by adaptive filtering. Then, to model users' multi-scale intentions, MFTSRec designs a multi-scale linear transformer and a multi-scale intention evolution and fusion layer, respectively, which can effectively capture the multi-scale dynamics of user intention.

To sum up, this paper summarizes several contributions points as follows:

This work proposes a new framework, named MFTSRec, which reveals multi-scale user intention dynamics, and has lower computational complexity and more vital anti-noise ability.

This paper designs a multi-scale linear transformer that maintains an evolving pattern of relationship-aware user interactions through a low-rank matrix decomposition and a multi-scale self-attentive mechanism.

This paper conducted extensive experiments on four publicly available datasets to verify that MFTSRec is superior to various state-of-the-art Sequential Recommender systems. Model ablation and robustness analysis further show the advantages of this model.

2. Literature review

Hidasi et al. first applied recurrent neural networks to sequential recommendation systems to mine the semantic information of items by capturing the sequential evolutionary relationships between items in the user interaction history [8]. Tang et al. proposed a CNN-based sequential recommender that treats the parameters matrix of a sequence as an "image" and extracts sequence transition information by convolution operations [9]. In recent years, transformer-based architectures have achieved SOTA results in many areas, such as Object Detection, Natural Language Understanding [10, 11]. Kang et al. first applied the transformer architecture to the sequential recommendation and proposed a self-attentive framework called SASRec [12]. SASRec uses a multi-headed self-focus mechanism to capture the sequential behavior of users and achieve state-of-the-art performance on a variety of datasets.

Despite the good results achieved by the converter in sequential recommendations, there are still two major challenges that make it a tricky problem:

Sequence denoising: There is often much noise in the sequence of user interactions, which often originates from users' unintentional clicks, wrong interactions, and even malicious fakes that do not reflect users' preferences and intentions. Deep neural networks, although effective, degrade significantly in the face of noise and tend to over-fit noisy data [13, 14]. When the recorded sequence data contains noise, the deep neural network may focus too much on the noisy signal and ignore the valuable features, making it difficult to learn the original sequence pattern of the data. Transformer-based deep neural networks are more problematic at this point because the self-attentive mechanism focuses on all items of sequence modelling.

Multi-scale intention dynamics: In a real-world scenario, the user's intention may show a multi-scale change pattern over time. Due to the multi-scale change pattern of user intention, there are multiple intentions embedded in the sequence, and user intention changes over time in the sequence view. In the past, traditional sequence recommendation algorithms often could only capture user intention at a single position but not at multiple consecutive positions. How to model the multi-scale change pattern of user intention becomes a major challenge for sequence recommendation.

3. Preliminary

The sequential recommendation system uses the user's past interaction behavior to predict the next interaction item, formally defined as follows. There are a set of users, denoted by U, whose size is denoted by $|U|$, and each element and each element u in the set U, represents a user. Similarly, there is an item set I, where each element $i \in I$ represents an item and the set size is $|I|$. Items mean different things in different scenarios. In the e-commerce recommendation system, the item represents

DOI: 10.54254/2755-2721/13/20230723

one commodity, while in the movie recommendation system, the item represents one movie. Each user interacts with multiple items, and a sequence of interactions $u^s = i_1, i_2, i_3, \dots, i_{|u^s|}$ can be constructed for each user u based on the time sequence in which each interaction occurs, where $|u^s|$ denotes the number of interactions sequence and $i_t (i_t \in I)$ is the t-th item that the user has interacted with. The sequential recommendation system predicts the next possible interaction i_{t+1} by capturing the dynamic preference preferences of the user. Its inputs and outputs are defined as follows:

Input: The interaction sequence for each user $i_1, i_2, i_3, \dots, i_{|u^s|}$.

Output: The possibility of the user interacting with the next candidate item i_{t+1} time t+1 steps.

4. Methodology

Figure 1 illustrates the overall architecture of MFTSRec under user-specified interaction sequences. Firstly, the embedding layer initializes all items and their corresponding sequence-position embeddings. Secondly, MFTSRec constructs a sequence signal filtering module to adaptively attenuate the noisy signals in the sequence and extract meaningful features from all frequencies. Thirdly, a multi-scale linear Transformer is introduced to capture the user's preferred multi-scale variation patterns in linear time complexity. Fourthly, by using an intention evolution and fusion layer, the multi-scale dynamic preference pattern is integrated into the common latent representation space. Finally, through a goal-aware prediction layer, items of interest to the user are recommended.

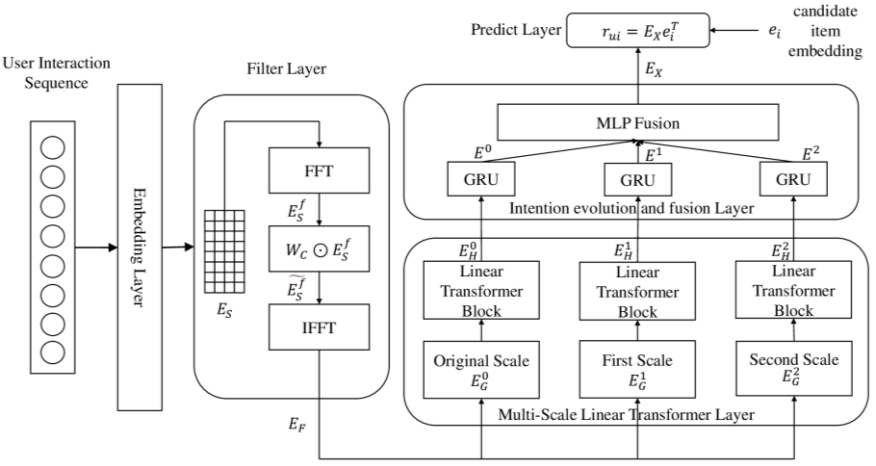


Figure 1. Overview of the proposed network.

4.1. Embedding layer

Firstly, the input sequence $i_1, i_2, i_3, \dots, i_{|u^s|}$ of the model is transformed into a fixed length sequence $i_1, i_2, i_3, \dots, i_{|L|}$, where L is the preset maximum length of the sequence. When the sequence length is less than L, MFTSRec fills the sequence with "0" until the sequence length is L; When the sequence length is greater than L, MFTSRec uses the sliding window method to decompose a long sequence into multiple sequences of length L by setting the window size as L and the step size as 1. Secondly, to represent the semantic and positional characteristics of the items, MFTSRec constructs a learnable item embedding matrix M_I and a learnable position embedding matrix M_P, respectively, where $M_I \in R^{|I| \times d}$, $M_P \in R^{L \times d}$, d is the latent dimensionality. By summing the two embedding matrices, the sequence representation E_S can be obtained as shown in Equation 1.

$$E = \begin{bmatrix} e_{i_1} + e_{p_1} \\ e_{i_2} + e_{p_2} \\ \vdots \\ e_{i_L} + e_{p_L} \end{bmatrix} \tag{1}$$

DOI: 10.54254/2755-2721/13/20230723

Where e_{i_t} is the dense vector corresponding to item i_t, e_{p_t} is the dense vector corresponding to the t-th position information, and $E\left(E \in R^{L\times d}\right)$ is the sequence representation after the Embedding layer. Since the embedding is generated by random initialization, it is prone to the problems of unstable training and overfitting [12]. Inspired by recent works, this paper uses dropout to mitigate overfitting and LayerNorm to make the training more stable [11, 15]. Thus, the embedding layer generates the final sequence representation $E_S\left(E_S \in R^{L\times d}\right)$ by Equation 2.

$$E_S = Dropout\left(Layernorm(E)\right) \tag{2}$$

4.2. Filter layer

Inspired by denoising filtering algorithms in the field of digital signal processing, a filtering layer is customized to solve the sequence noise problem [16]. Filter layer first performs a discrete Fourier transform on each dimension of the sequence data representation obtained from the embedding layer to obtain a frequency domain representation of the sequence features, as shown in Equation 3.

$$E_S^f = \mathcal{F}(E_S) \in C^{L\times d} \tag{3}$$

Where $\mathcal{F}(\cdot)$ represents the Fourier transform of one dimension and C represents the complex domain. E_S^f is a complex tensor representing the sequence's frequency domain characteristics. The filter layer then constructs a learnable filter that encodes the input sequential signals in the frequency domain and optimizes it directly from the original embedding, as shown in Equation 4.

$$E_s^f = W_C \odot E_s^f \tag{4}$$

Where $W_C \in C^{L\times d}$ is a learnable complex parameter, and $\odot$ is the Hadamard product. Since W_C can be optimized by the stochastic gradient descent algorithm (SGD), our filter can adaptively represent any filter in the frequency domain [17]. By multiplying by a learnable filter, our filtering layer can automatically adjust the spectrum to remove the noise information from the sequence to obtain a valuable feature representation. Finally, Filter layer uses the inverse FFT transform to convert E_s^f from the frequency domain to the time domain, and can extract sufficient sequence information due to a larger field of perception.

$$E_F = \mathcal{F}^{-1}\left(E_s^f\right) \in R^{L\times d} \tag{5}$$

Where $\mathcal{F}^{-1}(\cdot)$ stands for one-dimensional Fourier inverse transform, which can convert the complex tensor features in the frequency domain to real tensor features in the time domain. With the Fourier transform and Fourier inverse transform, MFTSRec can effectively remove the noise information from the sequence, thus providing a more qualitative sequence characterization for the subsequent modules.

4.3. Multi-scale linear transformer layer

To capture multiple intention representations in sequence views and effectively reduce computational efficiency, MFTSRec proposes a simple multi-scale linear Transformer module that effectively captures multiple point-level intention representations and scale-level intention representations of users and significantly reduces the time complexity of the Transformer, thus enabling easier training.

4.3.1. Linear self-attention module. The Transformer is a model consisting of multiple self-attentive layers. The self-attentive mechanism in Transformer enables the model to systematically rank the different components of the input data and then perform information aggregation on these data, allowing the model to focus on the role of individual input features. However, the self-attentive mechanism requires the model to compute the similarity of any two positional representations in the sequence, thus reaching a time complexity of $O(L^2 d)$, where L indicates the length of the sequence and d denotes the dimensionality of the potential representation. When the sequence length is too long, the

DOI: 10.54254/2755-2721/13/20230723

computational cost of the self-attentive mechanism grows dramatically, limiting its application in practical scenarios [18]. MFTSRec uses a low-rank decomposition technique to address this challenge to enhance the transformer's transformation operation, which greatly reduces the original time complexity.

In contrast to the original two-two vector inner product calculation of attention operations, MFTSRec approximates the original attention by generating multiple smaller attention operations via a low-rank decomposition technique. First, MFTSRec defines two low-rank projection matrices, $P_r^1 \in R^{L\times r}$ and $P_r^2 \in R^{L\times r}$, where L denotes the length of the original sequence and r is the low-rank order of the projection, with r much smaller than L. Since the original attention matrix is implemented via a multi-headed mechanism, the final attention matrix has a rank less than or equal to d_h, d_h being the embedding dimension of each head in the multi-headed attention mechanism. Therefore, MFTSRec defines L as greater than or equal to d_h to keep the expressiveness of the attention, as shown in Equation 6.

$$E_m^l = \text{Softmax}\left(\frac{E_H^{l-1}\cdot W_m^Q\cdot(P_r^1\cdot E_H^{l-1}\cdot W_m^K)^T}{\sqrt{\frac{d}{h}}}\right)\cdot P_r^2\cdot E_H^{l-1}\cdot W_m^V \tag{6}$$

Where W_m^Q, W_m^K and W_m^V are learnable parameters ($W_m^Q, W_m^K, W_m^V \in R^{d\times d_h}$). Consistent with past work, MFTSRec maps the original d-dimensional vector into a d_h-dimensional vector by three linear projection matrices, where m is the number corresponding to each head in the multi-headed self-attentive mechanism and d is the dimension of the potential embedding, h is the total number of heads [11,15]. P_r^1 and P_r^2 are low-rank projection matrices ($P_r^1, P_r^2 \in R^{L\times J}$) that project the original matrices of length L into the space of low-rank potential representations of number J.

In summary, by performing low-rank factorization of the original matrix inner product operation, MFTSRec computes the self-attention matrix $M = \frac{E_H^{l-1}\cdot W_m^Q\cdot(P_r^1\cdot E_H^{l-1}\cdot W_m^K)^T}{\sqrt{\frac{d}{h}}}$ by size $R^{L\times J}$ compared to the original size $R^{L\times L}$. With this, the computational cost of the linear converter encoder can vary from from $O(L^2d)$ to $O(LJd)$, considering the low-rank projection dimension J, which is usually much smaller than L. Then MFTSRec performs the concatenation operation as well as the linear projection to obtain the representation after self-attention, as shown in Equation 7.

$$E_P^l = W_P[E_1^l\|\cdots\|E_m^l\|\cdots\|E_h^l] \tag{7}$$

Where $\|$ stands for concatenation operation and $W_p \in R^{d\times d}$ is the linear projection matrix. After performing multi-head vector Concatenation and linear projection, MFTSRec performs Dropout, Residual connection, LayerNorm, and a Feed-forward neural network on E_P^l to get a representation of the next layer, as shown in Equation 8 [15, 19-21].

$$E_H^l = FeedForward(Layernorm(Dropout(E_P^l) + E_H^{l-1})) \tag{8}$$

4.3.2. Multi-scale transformer. To enable the MFTSRec model to learn the intention dynamics at multiple sequential positions efficiently, MFTSRec proposes to augment our linear Transformer module with a multilevel structure to capture scale-specific intention dependence relationships. Specifically, MFTSRec customizes a multi-scale intention-aware generator to generate scale-specific representations E_G^p to maintain intention dynamics within scales. Here, MFTSRec defines a function $g(\cdot)$ as a multi-scale intention sequence extraction, and p is an intention scale, where MFTSRec constructs a scale-specific intention sequence as shown in Equation 9.

DOI: 10.54254/2755-2721/13/20230723

$$E_G^p = g(E_F, p) = \begin{bmatrix} Aggre(E_F^{0\times p+1}, E_F^{0\times p+2}, \dots E_F^{0\times p+p}) \\ Aggre(E_F^{1\times p+1}, E_F^{1\times p+2}, \dots E_F^{1\times p+p}) \\ \vdots \\ Aggre(E_F^{k\times P+1}, E_F^{k\times P+2}, \dots E_F^{k\times p+p}) \end{bmatrix} \quad (9)$$

Where p is defined as the subsequence length at a specific scale, E_F is from the Filter layer. E_F^k is the representation at a specific position in E_F and is a one-dimensional vector ($E_F^k \in R^d$). $Aggre(\cdot)$ is the aggregator that aggregates multi-scale interest. Here, MFTSRec uses the average pool to perform the embedding aggregation, about which other aggregation approaches can be left for future studies. By aggregating subsequences at a specific scale, this module can extract the once-varying dynamics $E_G^p \in R^{\frac{L}{p}\times d}$ at that scale. After that, this module feed the scale-specific sequence representations into the Linear Transformer layer for encoding scale-specific sequence patterns as shown below.

$$E_H^p = TransformerEncoder(E_G^p) \quad (10)$$

E_H^p denotes the encoding of short-term transition patterns for subsequences of scale p. In our MFTSRec framework, this module constructs two different scales (p_1 and p_2) to enhance our sequence representation learning so that our encoder can generate three scale-specific sequence embeddings $\widehat{E_H} \in R^{L\times d}$, $E_H^{p_1} \in R^{\frac{L}{p_1}\times d}$, $E_H^{p_2} \in R^{\frac{L}{p_2}\times d}$.

4.4. Intention evolution and fusion layer

4.4.1. Intention evolution layer. Users' intention is to exist at multiple granularities. At the coarse-grained scale, users may be temporarily interested in various movies at one time and need books at another; at the fine-grained scale, the evolution of users' intention is reflected in the change of individual specific goods. However, using only the above multi-scale Transformer layer without considering the evolution between core intentions, it is not enough to consider temporal information using positional embeddings only, which will undoubtedly lead to temporal order bias. For the final presentation of intentions to provide more relevant historical information, the temporal relationship between intentions also needs to be considered.

Thanks to multi-scale intention extraction, the intention embedding matrix at different scales maintain the temporal order of user intention. Intuitively, MFTSRec can use time series models to simulate the evolution of focused intentions. This paper uses a single sequence model to simulate the evolution of intentions., as in Equation 11.

$$E^{p_k} = GRU(E_H^{p_k}) \quad (11)$$

Where p_k is the representation at the k-th scale. Note that the evolution of intention at different scales yields a final representation of intention, so the original three matrices, $\widehat{E_H}$, $E_H^{p_1}$, and $E_H^{p_2}$, become three vectors $\hat{E} \in R^d$, $E^{p_1} \in R^d$, $E^{p_2} \in R^d$.

4.4.2. Intention fusion layer. To integrate multi-scale intention evolution representations into the same latent representation space, MFTSRec proposes to aggregate the scale-specific vectors with the following functions:

$$E_X = f(\hat{E} \| E^{p_1} \| E^{p_2}) \quad (12)$$

Here E_X is a vector, and $f(\cdot)$ denotes a projection function that projects a vector of dimension R^{3d} into the space of dimension R^d. $\|$ denotes a concatenation operation on different embedding vectors.

4.5. Prediction and training

DOI: 10.54254/2755-2721/13/20230723

4.5.1. Prediction. We evaluate the user's preferences to the target item by vector inner product and employ a shared item embedding strategy (As shown in Equation 13) [12, 22-23].

$$r_{ui} = E_X {e_i}^T \tag{13}$$

where E_X is the final representation of the user at multi-scale intention, e_i is the embedding of item i, and r_{ui} is the possibility score for user u to interact with item i.

4.5.2. Training. This paper used the BPR loss function to train the model, a pairwise approach for personalized ranking [24]. Expressly, BPR assumes that observed interactions better reflect users' preferences and should be assigned higher predictive values than unobserved ones (As shown in Equation 14).

$$Loss = \sum_{(u,i,j)\in O} - ln\, \sigma\,(r_{ui} - r_{uj}) + \lambda \|\theta\|_2^2 \tag{14}$$

Where O denotes paired training data, and for each user u, whose next interacted authentic item is i, MFTSRec samples a negative sample from that user's unobserved item j and construct the BPR loss through positive and negative sample pairs. $\sigma(\cdot)$is the sigmoid function, and λ controls the strength of L2 regularization to prevent overfitting. We use the Adam optimizer to optimize the model [25].

5. Experiments

This paper answers the following research questions and conducts experimental analysis by conducting experiments on four real-world datasets:

RQ1: How does MFTSRec perform overall under each dataset compared to various baseline methods?

RQ2: How each major module of MFTSRec (e.g., filter layer, multi-scale linear transformer encoder, intention evolution and fusion Layer) contributes to the overall performance?

RQ3: How does MFTSRec perform against noise-added datasets when compared to state-of-the-art methods??

RQ4: Whether different hyperparameters in MFTSRec can affect its performance?

5.1. Experimental settings

5.1.1. Datasets. Four real-world datasets from two different application sites were used to evaluate the model. The sparsity between the domains of the data sets varies greatly, as detailed in table 1.

Table 1. Statistics of the datasets after preprocessing.

Dataset	Users	Items	Interactions	Sparsity
MovieLens-1M	6040	3416	999611	95.16%
Amazon Instant Video	5130	1685	37126	99.57%
Amazon Beauty	22363	12101	198502	99.93%
Amazon Toys And Games	19413	11924	167597	99.93%

MovieLens: The MovieLens dataset contains rating data of multiple movies by multiple users, as well as movie metadata information and user attribute information. This dataset is often used as a recommendation system, a test dataset for machine learning algorithms [26]. The version we use (MovieLens-1M) includes 1 million user ratings, and we ignore the rating data, user attribute information, and movie metadata.

Amazon: Amazon is a large e-commerce platform that offers a range of datasets for recommendations based on the type of product [27]. Compared to the MovieLens dataset, the Amazon dataset is sparser and therefore is affected by noise to a greater extent. In this thesis, three categories of the Amazon dataset are considered, they are "Beauty," "Instant Video," and "Toys and Games." These three datasets are used more frequently for recommendation testing.

DOI: 10.54254/2755-2721/13/20230723

For these four datasets, this paper first removes users and items with too few interactions, which is consistent with past work [7, 12]. Next, this paper groups the datasets by user, and each group was sorted in ascending order by timestamp. For each interaction of each user, MFTSRec samples one item from the set of items that the user has never interacted with to form a team of positive and negative sample pairs.

5.1.2. Evaluation protocols. We adopt the "leave one out" evaluation strategy, as is commonly used in past work [12, 23]. Using the last two items in the user interaction sequence as the validation and test sets, this paper can evaluate how well the model predicts the user's upcoming interactions with the items. We evaluate all methods using widely used metrics, including $NDCG@K$ and $HR@K$[28]. $NDCG@K$ is used to evaluate the ranking accuracy, which increases as the model ranking output is closer to the actual ranking, and $HR@K$ is used to evaluate the percentage of recommended items in the items that users actually interact with, which does not consider the ranking of the items. The specific calculation is shown in Equation 15, 16.

$$NDCG@K = \frac{1}{Z} DCG@K = \frac{1}{Z} \sum_{i=1}^{K} \frac{2^{I(|T_u \cap \{r_{ui}\}|)} - 1}{\log_2(i+1)} \tag{15}$$

$$HR@K = \frac{1}{m} \sum_u I(|R_u \cap T_u|) \tag{16}$$

Where Z indicates a constant associated with K. T_u is the interaction item set in test for user u, and r_{ui} denotes the i-th item in recommend list R_u based on predict score. $I(\cdot)$ denotes a function that outputs 1 when the input is True and 0 otherwise. m is the number of users in the test set. Different from past work, this paper randomly samples 1000 negative samples per user for performance evaluation [29].

5.1.3. Baselines. To prove the validity of MFTSRec, this paper divides the baseline model into three groups. The first group does not consider sequence information and includes both non-personalized and personalized recommendation methods.

PopRec: This is a method of recommending items based on their popularity, counting, and ranking the frequency of each item and then giving recommendations.

BPRMF: A personalized pairwise loss function based on matrix decomposition is used for recommendation [24].

NCF: The method replaces the vector dot product operation in matrix factorization with a multilayer perceptron model to model complex nonlinear relationships in user-item interactions [5]. The second group includes some typical models based on the interaction order of items, these methods include convolutional neural networks, recurrent neural networks, and self-attentive methods.

GRU4Rec: This is a typical recurrent neural network-based method modeling item sequence evolution relationships, and it also employs gated recursive units to model the user's sequence interaction patterns and control the flow of information based on gating mechanisms [8].

Caser: This is a widely used baseline based on convolutional neural networks, which perform aggregation across time interaction by convolution in both directions [9].

SASRec: For the first time, Transformer is applied to a sequential recommender system to encode sequential patterns of user interactions through a self-attentive mechanism [12]. The third group includes sequential recommendation methods based on multi-intention modelling, which are relatively new and have primarily achieved the latest SOTA results.

DSSRec: The model proposes a sequence-to-sequence training approach by enhancing the consistency of the input sequence with multiple intentions in the output sequence [30].

SURGE: In SURGE, metric learning is used to model each interaction sequence into a graph structure. In using attention mechanisms and graph pooling techniques, multiple intentions in the graph are aggregated into core intentions with different signal strengths as well as edge intentions. Finally, the

DOI: 10.54254/2755-2721/13/20230723

evolution of intentions is modelled by a sequence web, and the model has achieved the SOTA effect in recent years [31].

5.1.4. Hyperparameter settings. The training phase used a learning rate of 0.001 with 0.96 exponential decay. The training batch size is chosen from [64, 128, 256, 512, 1024], and the hidden layer dimensions are chosen from [8, 16, 32, 64, 128]. The maximum length of the sequence is set to 50. L2 regularization coefficients are chosen from [0.001, 0.005, 0.01, 0.05]. For the baseline model, we used the code provided by the original authors or implemented strictly according to the original paper.

5.2. Performance comparison (RQ1)

Table 2. Overall model performance on four datasets, with the metrics of HR@K and NDCG@K(K=10).

Model	MoviesLens-1M		Beauty		Instant Video		Toys and Games	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
General Recommendation Methods								
PopRec	0.0700	0.0320	0.0550	0.0276	0.0156	0.0074	0.0515	0.0255
BPRMF	0.1113	0.0493	0.1376	0.0754	0.1957	0.0896	0.1210	0.0659
NCF	0.1083	0.0500	0.1297	0.0696	0.1908	0.0897	0.1063	0.0567
Transitional Sequential Recommendation System								
GRU4Rec	0.4709	0.2803	0.2154	0.1294	0.2600	0.1675	0.1943	0.1140
Caser	0.4341	0.2538	0.1744	0.0969	0.2117	0.1203	0.1390	0.0772
SASRec	0.4836	0.2914	0.2511	0.1569	0.3072	0.1893	0.2447	0.1561
Multi-Intention Sequential Recommendation System								
DSSRec	<u>0.5217</u>	0.3047	0.2911	<u>0.1862</u>	0.3729	0.2306	0.2853	0.1858
SURGE	0.5132	<u>0.3233</u>	<u>0.3101</u>	0.1794	<u>0.3941</u>	<u>0.2408</u>	<u>0.2912</u>	<u>0.1923</u>
MFTSRec	0.5543	0.3407	0.3528	0.1989	0.4321	0.2714	0.3274	0.2148
#Improve	6.25%	5.38%	13.77%	10.87%	9.64%	12.71%	12.43%	11.70%

A detailed performance comparison of different methods is reported in Table 2, and the summary of the results is as follows:

5.2.1. The effectiveness of MFTSRec model. Table 2 shows that MFTSRec outperforms all baseline models on the four real datasets. The average improvement rates of $Hit@10$ and $NDCG@10$ of MFTSRec over the best baseline model on the Movieslens-1M dataset were 6.25% and 5.38%, 6.67% and 6.82% on the Beauty dataset, 9.64% and 12.71% on Instant Video dataset, 6.94% and 11.70% on the Toys and Games dataset, which proves the effectiveness of MFTSRec. The performance improvement can be attributed to: i) Through the filter layer, our model can attenuate the noise present in the data, enabling the model to ignore the noise and thus capture the primary intention of the users; ii) Our multi-scale Transformer can capture the transitional patterns of user interaction sequences from coarse-grained to fine-grained.

5.2.2. Sequence information and multi-intention modelling both help improve model performance. Comparing the performance of each model, this paper finds that sequence information and multi-intention modelling lead to improved performance. GRU4Rec, Caser, and SASRec perform much better than NCF and BPRMF on all four datasets. This illustrates the importance of sequential patterns. The performance of SASRec has a good improvement over GRU4Rec and Caser, which indicates that the self-attentive mechanism provides powerful sequence encoding capabilities for better capturing the

DOI: 10.54254/2755-2721/13/20230723

underlying user intention in user interaction sequences. The performance of both DSSRec and SURGE is better than the traditional sequence recommendation model because DSSRec and SURGE are designed with complex multi-intention extraction modules, and SURGE utilizes the aggregation propagation mechanism of graph neural networks to capture the underlying core intention more effectively.

5.2.3. MFTSRec consistently outperforms multi-intention sequence recommendation models. When MFTSRec is compared to other multi-intention sequential recommendation models, performance is significantly improved. This performance improvement is attributed to the denoising of user interaction sequences and multi-scale pattern modelling. For example, DSSRec defines a fixed number of intention prototypes, which cannot effectively capture the intention information in the user interaction sequence. SURGE follows the paradigm of graph attention networks, which are susceptible to noise and overfitting. In addition, neither approach focuses on the multi-scale patterns in user interaction sequences, which can reflect the changing process of user intentions at different granularities.

5.3. Ablation study (RQ2)

This section performs ablation experiments on three important modules of MFTSRec, each corresponding to one modeling consideration. The design of the ablation model is shown below:

MFTSRec w/o Filter: The Filter layer is removed, and this model variable is susceptible to the noise items in the sequence.

MFTSRec w/o MST: The Multi-Scale Linear Transformer is removed, and this paper simply utilizes a vanilla Transformer instead [12]. This model cannot extract multi-scale intention from the interaction sequence.

MFTSRec w/o IEF: The Intention evolution and fusion is removed, this module cannot extract the evolution of the user's intention at each scale over time, and this paper aggregates the intention representations using a mean value approach.

Table 3. The ablation study of MFTSRec on Instant Video Dataset, with the metrics of HR@K and NDCG@K (K=5,10).

Model	Instant Video			
	HR@5	NDCG@5	HR@10	NDCG@10
MFTSRec w/o Filter	0.3217	0.2311	0.4122	0.2558
MFTSRec w/o MST	0.3015	0.2264	0.4025	0.2511
MFTSRec w/o IEF	0.3114	0.2293	0.4087	0.2533
MFTSRec	0.3421	0.2522	0.4321	0.2714

The ablation study results are shown in table 3. From the evaluation results, this paper draws the following conclusions:

The performance gap between MFTSRec and w/o Filter shows that the learnable filter MFTSRec designed can mitigate the interference of noisy information, which helps to better learn the real data distribution and better learn sequential patterns when dealing with data containing noise.

The performance gap between MFTSRec and w/o MST shows that the multi-Scale intention extraction this paper designed effectively captures users' behavioral dynamics at different periods, resulting in performance gains for the model. This observation justifies our ability to enhance Transformer representation using multi-scale relationships within sequences.

The performance gap between MFTSRec and w/o IEF suggests that the multi-scale intentional evolution and aggregation layer this paper designed can compensate for the lack of the Transformer's ability to model sequence position information and thus capture the evolutionary relationships between sequences.

5.4. Robustness analysis (RQ3)

DOI: 10.54254/2755-2721/13/20230723

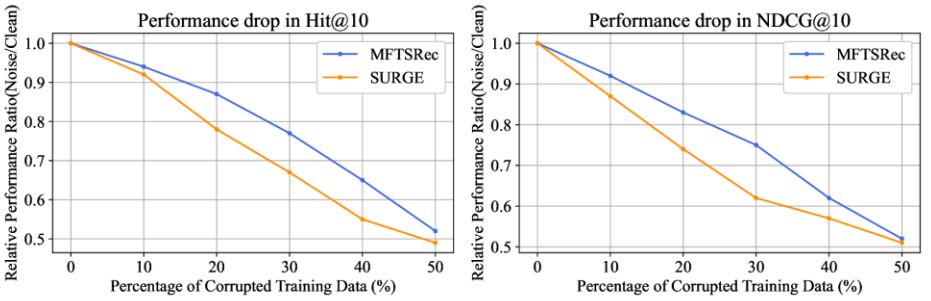
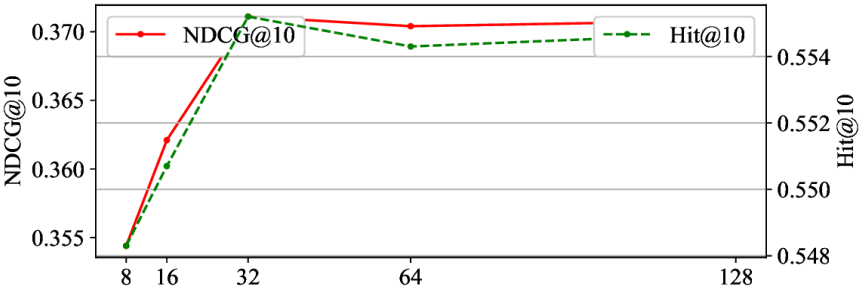


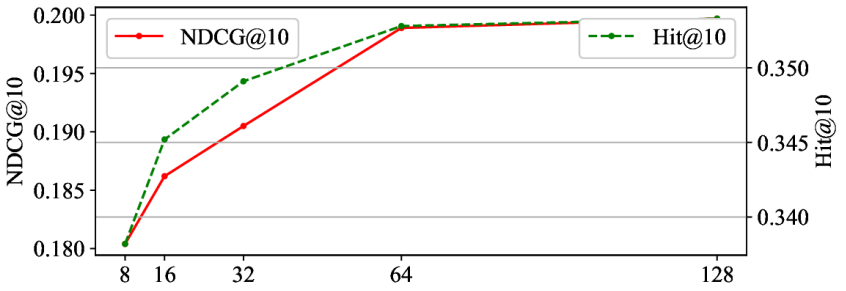
Figure 2. The relative performance ratio of the dataset instant video when the data is corrupted by random noise. y-axis is the ratio of performance with noisy training data to performance with clean training data, and x-axis is the proportion of noise added.

We now analyze the robustness of our MFTSRec model compared to the best baseline model, SURGE. Specifically, in this paper, noise is added to each user's interaction sequence on the instant video dataset by random insertion and replacement operations, with the percentage of noise added ranging from 10% to 50%. We show the performance degradation of both methods, MFTSRec as well as SURGE, in figure 2. We can see that the performance degradation of MFTSRec is lower when the percentage of corrupted data is below 10%, while the performance degradation of SURGE is always larger than that of MFTSRec. This indicates that our MFTSRec does have the potential to lead to higher robustness by attenuating a certain amount of noise signal in the frequency domain and extracting multi-scale representations of intention.

5.5. Hyper-parameter sensitivity (RQ4)



(a) MovieLens-1M



(b) Beauty

Figure 3. Effect of the latent dimensionality D.

DOI: 10.54254/2755-2721/13/20230723

MFTSRec has a key hyperparameter, i.e., the embedding size D. To investigate the effect of different embedding dimension settings on MFTSRec, this paper conducted experiments on the MovieLens-1M and Beauty datasets with different configurations of the key hyperparameters. As shown in the figure, this paper draws the following conclusions: figure 3(a) and 3(b) show the set of the model when the hidden dimension d grows from 8 to 128. The performance of both MovieLens-1M and Beauty datasets improves as the latent dimension grows from 8 to 32 because higher latent dimensions can lead to a wide range of expressiveness. However, as the model dimensionality increases further from 32 to 64, the performance of MovieLens-1M saturates while that of Beauty improves. Notice that the model performance does not improve when the embedding dimension d continues to increase. First, increasing the embedding dimension increases the capacity of the model, but it is more likely to lead to overfitting. Second, as the embedding dimension increases, it introduces additional computational overhead and memory overhead. In addition, we also note that the optimal embedding dimension varies for different datasets. For example, the Beauty itemset is 4 times larger than MovieLens-1M, so the Beauty dataset needs a higher potential dimension to achieve better performance.

6. Conclusion

This paper proposes a new sequential recommendation model MFTSRec for the next item recommendation. MFTSRec combines frequency domain filtering, low-rank decomposition, and multi-scale decomposition to alleviate the noise problem in the original sequence and effectively extracts the interaction dynamics of users at multiple scales. Extensive experiments on four real-world datasets demonstrate the effectiveness of MFTSRec. Further studies of ablation experiments and robustness analysis confirm that our approach provides more effective and stable recommendations than other baseline models. In the future, we will introduce user portraits and item attribute information to model user interaction sequences at a more granular level.

References

[1] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, et al. In Alexandros Karat- zoglou, Balázs Hidasi, Domonkos Tikk, Oren Sar Shalom, Haggai Roitman, Bracha Shapira, and Lior Rokach, editors, Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS@RecSys 2016, Boston, MA, USA, September 15, 2016, pages 7–10. ACM. 2016.

[2] Huifeng Guo, Ruiming Tang, et al. Deepfm: A factorization-machine based neural network for CTR prediction. In Carles Sierra, editor, Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, pages 1725– 1731. ijcai.org. 2017.

[3] James Davidson, Benjamin Liebald, Junning Liu, et al. The youtube video recommendation system. In Xavier Amatriain, Marc Torrens, Paul Resnick, and Markus Zanker, editors, Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010, pages 293–296. ACM. 2010.

[4] Aäron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content- based music recommendation. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pages 2643–2651. 2013.

[5] Xiangnan He, Lizi Liao, et al. Neural collaborative filtering. In Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich, editors, Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017, pages 173–182. ACM. 2017.

[6] Xiang Wang, Xiangnan He, et al. Neural graph collaborative filtering. In Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer, editors, Proceedings of the 42nd International ACM SIGIR Conference on Research and Development

DOI: 10.54254/2755-2721/13/20230723

in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019, pages 165– 174. ACM. 2019.

[7] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010, pages 811–820. ACM. 2010.

[8] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939. 2015.

[9] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In Proceedings of the eleventh ACM international conference on web search and data mining, pages 565–573. 2018.

[10] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End- to-end object detection with transformers. In European conference on computer vision, pages 213–229. Springer. 2020.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre- training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[12] Kang, W. C., & McAuley, J. Self-attentive sequential recommendation. In 2018 IEEE international conference on data mining (ICDM) (pp. 197-206). IEEE. (2018, November).

[13] J. Lever, M. Krzywinski, and N. Altman. Points of significance: Model selection and overfitting. Nature Methods, 13(9):703–704. 2016.

[14] Rich Caruana, Steve Lawrence, and C. Lee Giles. Overfitting in neural nets: Back- propagation, conjugate gradient, and early stopping. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA, pages 402–408. MIT Press. 2000.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. Advances in neural information processing systems. 2017.

[16] C. K. Yuen. Review of “theory and application of digital signal processing” by lawrence r. rabiner and bernard gold. IEEE Trans. Syst. Man Cybern., 8(2):146. 1978.

[17] H. Robbins and S. Monro. A stochastic approximation method. Annals of Mathematical Statistics, 22(3):400–407. 1951.

[18] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient trans- former. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net. 2020.

[19] Nitish Srivastava, Geoffrey Hinton, et al. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1):1929–1958. 2014.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778. 2016.

[21] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. arXiv preprint arXiv:1607.06450. 2016.

[22] Jiacheng Li, Yujie Wang, and Julian McAuley. Time interval aware self-attention for sequential recommendation. In Proceedings of the 13th international conference on web search and data mining, pages 322–330. 2020.

[23] Fei Sun, Jun Liu, Jian Wu, et al. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In Proceedings of the 28th ACM international conference on information and knowledge management, pages 1441–1450. 2019.

[24] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In Jeff A. Bilmes and Andrew Y. Ng,

DOI: 10.54254/2755-2721/13/20230723

editors, UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009, pages 452–461. AUAI Press. 2009.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014.

[26] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis), 5(4):1–19. 2015.

[27] Julian McAuley, Christopher Targett et al. Image-based recommendations on styles and substitutes. In Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, pages 43–52. 2015.

[28] Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. SIGIR Forum, 51(2):243–250. 2017.

[29] Koren, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 426-434). (2008, August).

[30] Jianxin Ma, Chang Zhou, Hongxia Yang, et al. Disentangled self-supervision in sequential recommenders. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 483–491. 2020.

[31] Jianxin Chang, Chen Gao, Yu Zheng, et al. Sequential recommendation with graph neural networks. In Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai, editors, SIGIR ’21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021, pages 378–387. ACM. 2021.