



Multi-temporal Sequential Recommendation Model Based on the Fused Learning Preferences

Jianxia Chen¹ · Liwei Pan¹ · Shi Dong² · Tianci Yu¹ · Liang Xiao¹ · Meihan Yao¹ · Shijie Luo¹

Received: 1 May 2023 / Accepted: 31 July 2023
© The Author(s) 2023

Abstract

Since sequential recommendation can effectively capture the user's dynamic interaction pattern according to the user's historical interaction, it plays one of the important roles in the field of recommendation systems. Nowadays, the sequential recommendation models based on deep learning approaches have achieved good results in reality due to their excellent performance in information retrieval and filtering. However, there are still some challenges that still should be overcome of them. A major disadvantage of them is that they are only considering users' short-time interests while ignoring their long-term preferences. Moreover, they are incapable of considering the influence of time information in the original interaction sequence, which could be helpful to fully extract various patterns via the different temporal embedding forms. Therefore, this paper proposes a novel model named multi-temporal sequential recommendation model based on the fused learning preference named MTSR-FLP in short. In the proposed framework, this paper adopts an MLP-based state generation module to consider the user's long-term preferences and the short-time interests simultaneously. In particular, the proposed MTSR-FLP model designs a global representation learning approach to obtain the user's global preference, and a local representation learning approach to capture the users' local preference via its historical information. Moreover, the proposed model develops a multiple temporal embedding scheme to encode the positions of user-item interactions of a sequence, in which multiple kernels are utilized for the absolute or relative timestamps to establish unique embedding matrices. Finally, compared with other advanced sequence recommendation models on five public real-world datasets, the experimental results show that the proposed MTSR-FLP model has improved the performance of HR@10 from the 6.68% through 31.10% and NDCG@10 from the 8.60% through 42.54%.

Keywords Sequential recommendation · Deep learning · Time series embedding · Global representation learning · Local representation learning

1 Introduction

A recommendation system is an effective and powerful tool to address information overload by making users easily capture information in terms of their preferences and allowing online platforms to widely publicize the information they produce. Different from traditional recommendation systems, such as content-based and collaborative filtering systems, which prefer focusing on the user's static historical

interactions and assume that all user-item interactions in the historical sequence are equally important, sequential recommendation (SR) systems aimed at catching the dynamic preferences of users from their historical interaction. Therefore, SR is more consistent with real-world situations to obtain more accurate results with more information considered.

Generally, the conventional SR methods are classified into traditional machine learning approaches and deep learning-based approaches. The traditional machine learning (ML) approaches are mainly exploited to build up the sequence data, such as k-means, Markov chains, and matrix decomposition [1–4]. Especially, matrix factorization (MF) can obtain users' long-term preferences through various sequences [5]. Nowadays, the sequential recommendation models based on the deep learning approaches have achieved relatively good results in reality due to their

✉ Shi Dong
njbsok@163.com

¹ School of Computer Science, Hubei University of Technology, Wuhan 430068, China

² School of Computer Science and Technology, Zhoukou Normal University, Zhoukou 466001, China

excellent performance in information retrieval and filtering, mainly including convolutional neural networks (CNNs) [6] and recurrent neural networks (RNNs) [7]. This is because DL-based SR approaches utilize a much longer sequence which is beneficial to the semantic information of the whole sequence [8]. In addition, DL-based SR approaches also have achieved much better performance for both the sparse data and different length sequences.

However, there are still some challenges that still should be overcome in the above DL-based SR approaches. A major disadvantage of them is that they are only considering users' short-time interests while ignoring their long-term preferences. Moreover, they are incapable of considering the influence of time information in the original interaction sequence, which could have been helpful to fully extract various patterns by finding different temporal embedding forms. Therefore, this paper aims to propose a deep learning-based sequential recommendation system. It utilizes both the global preference and the local preference of users with the multi-temporal embedding to encode both absolute and relative positions of a user-item interaction, and simulates the real-time recommendation environment.

This paper proposes a novel model named multi-temporal sequential recommendation model based on the fused learning preference named MTSR-FLP. The proposed MTSR-FLP model consists of four parts: multi-temporal embedding learning; local representation learning; global representation learning; item similarity gating module. First, the multi-temporal embedding learning module is the most important and innovative in the proposed model. It is well designed especially to address the timestamp information with the construction of the multiple temporal embedding, which will be sent to the local representation learning module, named LRL in short. Moreover, the local representation learning module is developed via the framework of the SASRec model [9], which infers the multiple temporal embedding with the different weight and their aggregations, to obtain the local preference of users. Afterward, in the global representation learning module, named GRL in short, a novel location-based attention layer has been developed to obtain the final global learning representation of all users. Finally, the proposed model designs a novel item similarity gating module, which is critical in coordinating the local preference of the specific user and the global preference of all users. Meanwhile, it considers the influence of candidate items by calculating the similarity of candidate items and recently interacted items.

The main contributions of the proposed MTSR-FLP model are described as follows: (1) the proposed model develops a multiple temporal embedding scheme to encode the positions of user-item interactions of a sequence; (2) to capture the local representation of a user via his historical information, the MTSR-FLP model proposes a local

representation learning module; (3) to effectively capture the user's global preference, the proposed model designs a global representation learning module; (4) compared with the four traditional models on five public datasets, experimental results demonstrated that the proposed model had better performance regarding the effectiveness and feasibility of the sequence recommendation.

The rest of this paper is organized as follows. Section 2 briefly introduces the recent development of sequential recommendation based on deep learning and temporal recommendation. Section 3 describes the specific details of the proposed model and the procedures of its training. Experimental results on five public datasets and comparison with existing models are analyzed in Sect. 4. Finally, Sect. 5 gives a conclusion of the paper study and the directions for future work.

2 Related Work

2.1 DL-Based Sequential Recommendation

Recently, many researchers utilize DL-based sequential recommendation (SR) models to obtain nonlinear and dynamic features. In particular, DL models based on the RNNs are nearly the first to be incorporated into sequential modeling, which naturally models sequences step-by-step [10, 11]. Moreover, GRU4Rec model [12] has been proposed to effectively extract the position information between adjacent sequences, and GRU4Rec+ model [13] adopts various sampling strategies and loss functions. However, these models are more dependent on the user's recent interactions, weakening the training efficiency. Afterward, some researchers utilized CNNs-based models to address the vanishing gradient issue resulting from the RNNs-base SR models. For example, Caser model [5] learns sequence patterns using convolutional filters on the local sequences with the dense users' behavior data. A simple convolutional generative network is proposed to learn high-level representation from both short- and long-range item dependencies [14].

Since the attention mechanisms have an advantage in their proper hierarchical structure, some DL-based models utilize them to solve sequential recommendation problems. For example, as an entire and famous attention mechanism, the Transformer model [15] computes representations of its input and output without RNNs or CNNs, which makes the Transformer more parallelizable and faster to be trained with superior translation quality. However, the Transformer model does not directly utilize the position information in the sequence and cannot extract local features [16]. In addition, some SR models have been developed through the self-attention mechanism. For example, SASRec [9] not only can uncover the long-term user's preference but also

can predict the next recommended item mainly based on the recently interacted items. Bert4Rec [17] captures the correlation of items in both left-to-right and right-to-left directions to obtain a better sequence representation of user sequence. LSSA [18] deals with both long-term preferences and sequential dynamics based on a long and short-term self-attention network. However, they are limited to only capturing the sequential patterns between consecutive interactive items, ignoring the complex relationships between the high-order interactions.

Recently, graph neural networks (GNNs) have been widely utilized to model high-order transformation relationships in the area of SR. For example, NGCF [19] and LightGCN [20] argue that graph-based models can effectively deal with collaborative data and are critical in presenting user/item embedding. SRGNN [21] converts the sequence data into graph-structured data with the gated GNNs to perform message propagation on graphs. GCE-GNN [22] utilizes a GNN model for the session graph to learn item embedding with a session-aware attention mechanism. GC-MC [23] proposes a graph convolutional matrix completion (GC-MC) framework from the perspective of link prediction on graphs. However, the GC-MC framework only models the user's direct scoring of items in a convolutional layer. CTDNE [24] defines a temporal graph to learn the dynamic embedding of nodes. However, most of these GNN-based models only capture sequential transition patterns based on items' locations and identities, while ignoring the influence of contextual features such as time intervals, which will cause the models to fail to learn appropriate sequence representations.

Inspired by the SASRec [9], which has the advantages of not only uncovering the long-term user's preference but also predicting the next recommended item mainly based on the recently interacted items, the proposed model utilizes the SASRec as the framework of the local representation learning layer. Different from the SASRec, the proposed local representation learning module infers the multiple temporal embedding with the different weight and their aggregations, to improve the performance of the local preference of users. In addition, the paper has not utilized the GNNs so the proposed model has a disadvantage at the high-order transformation relationships in some areas of the SR.

2.2 Temporal Recommendation

Since user preferences are dynamic in the SR problems, it is essential to capture temporal information. Originally, matrix factorization (MF) has often been utilized in temporal recommendation systems due to its relatively high accuracy and scalability [25]. However, it is difficult for the rating prediction of new users via the MF. Collaborative filtering-based (CF) approaches also have been proven of success in

modeling temporal information [26]. However, some linear approaches are unsuitable for the more complex dynamics of temporal changes, and the transition matrix usually is homogeneous for all users [27]. In addition, it is impractical for large datasets due to their time-consuming performance and so on.

Afterward, there are also deep learning-based models in temporal recommendation systems, which encode temporal interval signals into the user preference representations to predict the user's future actions. For example, Time-LSTM [28] combined the LSTM model with several time gates to achieve better performance of the time intervals in users' interaction sequences. Zhang et al. [29] introduced a self-attention mechanism into the sequence-sense recommendation model that represented the user's temporal interest. CTA model [30] utilized multiple-parameter kernel functions to address the temporal information as an attention mechanism [31]. Tang and Wang [10] combined a convolutional sequence embedding model with Top-N sequential recommendation as a way for a temporal recommendation. TiSAS-Rec [32] is the absolute positions of items and the time intervals between them in a sequence. Moreover, TASER [33] proposes a temporal-aware network based on absolute and relative time patterns. KDA [34] utilizes the temporal decay functions of various relations between items to explore the time drift of their relation interactions.

In addition, some GNN-based temporal SR models have been proposed recently. For example, RetaGNN model [35] improves the inductive and transferable performance of the SR by encoding the sub-graphs resulting from the user-item pairs. Moreover, DSPR [36] proposes a novel sequential POI recommendation approach to capture the real-time impact and absolute time. DRL-SRe [37] obtains the dynamic preferences based on the global user-item graph for each time slice that deploys an auxiliary temporal prediction task over consecutive time slices. TARec [38] captures the user's dynamic preferences based on the combination of an adaptive search-based module and a time-aware module.

Inspired by TiSASRec model [32], this paper seeks to utilize multi-scale temporal information embeddings based on the absolute and relative time patterns to comprehensively capture the diverse patterns of users' behavior. However, self-attention operations and global preference learning have been integrated with the time-aware into the proposed model to improve the performance of the SR.

3 The MTSR-FLP Model

In this section, the paper proposes the MTSR-FLP model, i.e., fusing the global and the local preference of users with self-attention networks based on the multi-temporal embedding for the sequential recommendation. Without loss of

generality, the paper has a recommendation system with implicit feedback given by a set of users to a set of items. Therefore, the entire recommendation process can be defined as follows.

- Multiple temporal embeddings
 - Convert the $T = \{t_1; t_2; \dots; t_L\} \in \mathbb{R}^{|I|}$ into absolute embedding by combining $E^{Day} \in \mathbb{R}^{|I| \times d}$ and $E^{Pos} \in \mathbb{R}^{|I| \times d}$, which contain the information of the index and the time information of each item in sequence, respectively.
 - Convert the $T = \{t_1; t_2; \dots; t_L\} \in \mathbb{R}^{|I|}$ into relative embedding $E^{Sin} \in \mathbb{R}^{|I| \times |I| \times d}$, $E^{Exp} \in \mathbb{R}^{|I| \times |I| \times d}$ and $E^{Log} \in \mathbb{R}^{|I| \times |I| \times d}$, which present different temporal information to a greater extent.
- Local learning representation
 - Input matrix $X^{(0)} = \{x_1; x_2; \dots; x_L\} \in \mathbb{R}^{|I| \times d}$ is obtained, in which the absolute position embedding matrix $E^{Abs} = \{a_1; a_2; \dots; a_L\} \in \mathbb{R}^{|I| \times d}$ to the embedding item matrix $E \in \mathbb{R}^{|I| \times d}$.
 - Input the matrix $X^{(0)}$ into stacked self-attentive blocks (SAB) to get the output of the b^{th} block as the $X^{(b)} = SAB^{(b)}(X^{(b-1)})$ $b \in \{1, 2, \dots, B\}$.

- Global learning representation
 - The global representation of the sequences is formalized as $y = LBA(E) = \text{softmax}(q^s (EW'_k)^T) EW'_v$.
- Item similarity gating
 - The specific computing equation is formalized as

$$g = \sigma([m_{sl}, y, m_i] W_G + b_G),$$

where y represents the global representation, m_{sl} is the recently interacted item embedding vector, and m_i is the candidate item embedding vector. W_G and b_G are weight and bias separately.

3.1 MTSR-FLP Model Overview

As shown in Fig. 1, the MTSR-FLP model mainly consists of four layers as follows: multi-temporal embedding learning; local representation learning; global representation learning; item similarity gating. Moreover, the entire algorithm of the MTSR-FLP model is described in Algorithm 1. The roles of the four components of the proposed model are described as follows:

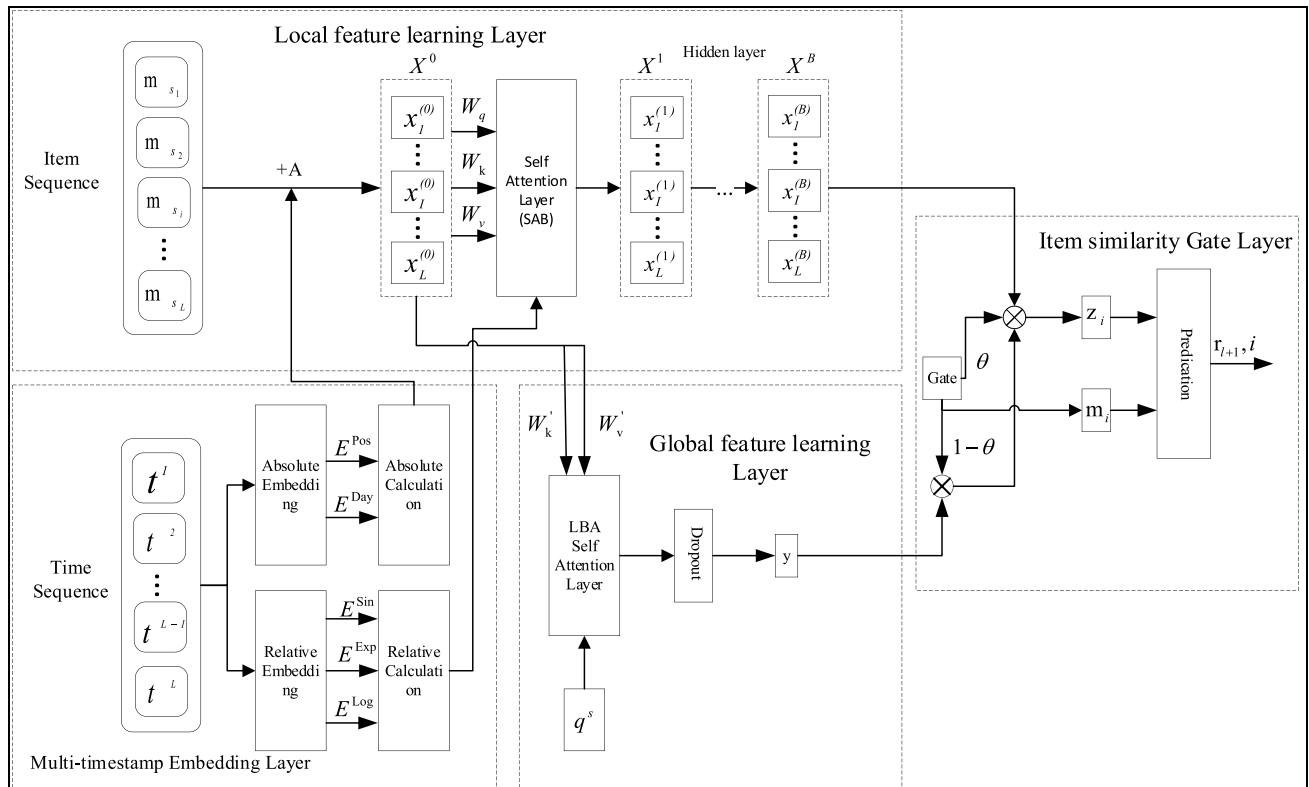


Fig. 1 The framework of the MTSR-FLP model

Algorithm 1 The MTSR-FLP algorithm

Input: The set of item datasets

Output: The set of predicted item datasets

1. Construct the multiple temporal embeddings: $E^{Exp}, E^{Log}, E^{Sin}, E^{Day}$ and E^{Pos}
 - 1.1 Convert time series embedding
 - 1.2 Calculate absolute embeddings and relative embeddings
2. Calculate local learning representation based on SASRec
 - 2.1 Add the absolute position embedding matrix $E^{Abs} = \{a_1; a_2; \dots; a_L\} \in \mathbb{R}^{||I|| \times d}$ to the embedding item matrix $E \in \mathbb{R}^{||I|| \times d}$ to obtain an input matrix $X^{(0)} = \{x_1; x_2; \dots; x_L\} \in \mathbb{R}^{||I|| \times d}$
 - 2.2 Input the matrix $X^{(0)}$ into stacked self-attentive blocks(SAB) to get the output of the b^{th} block
 - 2.3 Convert each SAB as a self-attentive layer(SAL), and then the results are fed into the feed forward layer.
3. Global representation learning
 - 3.1 Introduce $q^S \in \mathbb{R}^{1 \times d}$ as the query vector to calculate the most representative feature representation of all items in all sequences.
 - 3.2 Utilize the location-based attention layer(LBA) to obtain the global learning representation of all users: $y=LAB(E)$.
4. Item similarity gating
 - 4.1 Coordinate the local preference of the specific user and the global preference of all users as follows: $g = \sigma([m_{sl}, y, m_i]W_G + b_G)$
 - 4.2 Calculate the similarity of candidate items and recently interacted items as follows:

$$z_i = x_i \otimes g + y \otimes (1 - g)$$
 - 4.3 Calculate the final prediction of the item i preference to be the $(i + 1)^{th}$ item in the sequence

First, the multi-temporal embedding plays an important role in making full use of the temporal information of the user by constructing multiple temporal embeddings. In addition, the temporal embeddings are classified as two different forms of embedding vectors by the calculation way. Then, the paper utilizes a specific gate mechanism to integrate one form of embedding vectors as the absolute embeddings and integrate the other form of embedding vectors as the relative embeddings.

Second, the local representation learning layer mainly utilizes the self-attention mechanism, the absolute embeddings, and the relative embeddings learned in the multi-temporal embedding learning module to learn the local representation of the specific user.

Third, the global representation learning layer also utilizes all users' historical sequence to capture the global preference of all users using the attention mechanism which is modified based on the self-attention mechanism.

Finally, the item similarity gating layer balances the global representation and the local representation using gate mechanism.

As shown in Table 1, there are the symbolic representations of the diagram in Fig. 1.

3.2 Multi-temporal Embedding Learning Module

To address timestamp information, this module makes the following improvements:

- (1) The construction of multiple temporal embeddings: $E^{Exp}, E^{Log}, E^{Sin}, E^{Day}$, and E^{Pos} .
- (2) The construction of absolute embedding by combining E^{Day} and E^{Pos} .
- (3) The construction of relative embedding by combining E^{Exp}, E^{Log} , and E^{Sin} .

3.2.1 Time Series Embedding

In the user's historical sequence, each item is clicked by the user at a specific time, so time information is more important for recommendation.

Table 1 Notation

Notations	Description
S_u	Historical interaction sequence for user u
M^D	Day unit learnable embedding matrix
T	Timestamp sequence of user
$E^{Exp}, E^{Sin}, E^{Log}, E^{Day}$	Multiple temporal embedding vector
E^{Abs}	Absolute temporal embedding vector
E^{Rel}	Relative temporal embedding vector
W^Q, W^K, W^V	Embedding matrix of query, key and value
Δ	Unit lower triangular matrix
$r_{t+1,i}$	Output of model as item i

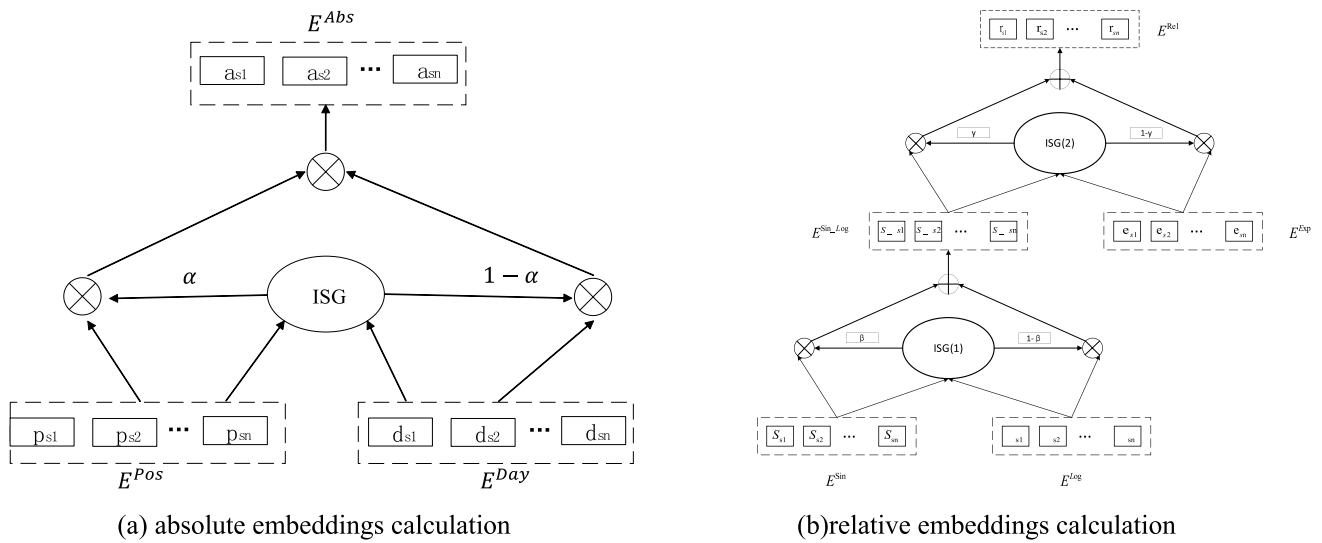


Fig. 2 Absolute and relative embeddings

The paper converts the $T = \{t_1; t_2; \dots; t_L\} \in \mathbb{R}^{|I|}$ into $E^{Day} \in \mathbb{R}^{|I| \times d}$ and $E^{Pos} \in \mathbb{R}^{|I| \times d}$. The d denotes the hidden vector dimension. E^{Pos} contains the information on the index of each item in a sequence. E^{Day} contains the time information of each item in a sequence. Compared with $E^{Timestamp}$, the E^{Day} is more reasonable. Since E^{Day} can save more storage space compared with $E^{Timestamp}$ using one day as a time interval, in the meantime it can take advantage of the time information of users' historical sequence.

In addition, the paper maintains a learnable embedding matrix $M^D \in \mathbb{R}^{|D| \times d}$, where $|D|$ is the time interval by taking a day as a unit of measure between the earliest clicked item and the latest clicked item in uses' historical sequence. Position embedding is a learnable positional embedding, where $M^P \in \mathbb{R}^{|I| \times d}$, where $|I|$ represents the length of the sequence. In this paper, the paper defines $|I|$ as 50.

The paper also converts the $T = \{t_1; t_2; \dots; t_L\} \in \mathbb{R}^{|I|}$ into $E^{Sin} \in \mathbb{R}^{|I| \times |I| \times d}$, $E^{Exp} \in \mathbb{R}^{|I| \times |I| \times d}$ and $E^{Log} \in \mathbb{R}^{|I| \times |I| \times d}$. This paper uses different time embeddings to utilize different temporal information to a greater extent. First, this paper defines a temporal differences matrix as $D \in \mathbb{R}^{N \times N}$, in which the element is defined as $d_{ab} = \frac{t_a - t_b}{\tau}$, where τ is the adjustable unit time difference; in the E^{Sin} , the paper converts the difference d_{ab} to a latent vector $\vec{\theta}_{ab} \in \mathbb{R}^{1 \times h}$, as shown in Eq. (1):

$$\vec{\theta}_{ab,2c} = \sin\left(\frac{d_{ab}}{\text{freq} \frac{2c}{h}}\right) \vec{\theta}_{ab,2c+1} = \cos\left(\frac{d_{ab}}{\text{freq} \frac{2c}{h}}\right) \quad (1)$$

where $\vec{\theta}_{ab,c}$ is the c^{th} value of the vector $\vec{\theta}_{ab}$. In the above equation, freq and h are adjustable parameters. Similarly, in the E^{Exp} , the paper converts d_{ab} to \vec{e}_{ab} and \vec{l}_{ab} respectively,

and the superposition of these vectors gives E^{Exp} and E^{Log} respectively, as shown in Eq. (2):

$$\vec{e}_{ab,c} = \exp\left(\frac{-|d_{ab}|}{\text{freq} \frac{c}{h}}\right) \vec{l}_{ab,c} = \log\left(1 + \frac{|d_{ab}|}{\text{freq} \frac{c}{h}}\right) \quad (2)$$

3.2.2 Absolute and Relative Embeddings Calculation

The calculation of absolute embeddings and relative embeddings is shown in Fig. 2. Here, E^{Day} and E^{Pos} are not simply addition operations to get the absolute embeddings, however, the paper calculates the different weights of the E^{Day} and E^{Pos} , then the paper adds them together via their weights. The specific calculation is described in Eq. (3):

$$\alpha = \sigma\left([E^{Day}, E^{Pos}] W_{\text{day_pos}} + b_{\text{day_pos}}\right) \quad (3)$$

where $[\cdot, \cdot]$ denotes the ternary concatenation operation, $W_{\text{day_pos}} \in \mathbb{R}^{2d \times 1}$ and $b_{\text{day_pos}} \in \mathbb{R}^{1 \times 1}$. To restrict each element in α from 0 to 1, the paper utilizes the sigmoid activation function $\sigma(z) = 1/(1 + e^{-z})$. Then, the paper adds E^{Day} to E^{Pos} to obtain E^{Abs} . The calculation procession is formalized in Eq. (4):

$$a_{s\ell} = \alpha_s \otimes p_{s\ell} + (1 - \alpha_s) \otimes p_{s\ell}, \ell \in \{1, 2, \dots, L\} \quad (4)$$

In the end, the paper can get the absolute embedding $E^{Abs} \in \mathbb{R}^{|I| \times d}$. Meanwhile, the paper calculates the relative embeddings with the same approaches of above. First, the paper can calculate β to balance the E^{Exp} and E^{Log} . This calculation process is described in Eqs. (5, 6):

$$\beta = \sigma([E^{\text{Sin}}, E^{\text{Log}}]W_{\text{sin_log}} + b_{\text{sin_log}}) \tag{5}$$

where $W_{\text{sin_log}} \in \mathbb{R}^{2d \times 1}$, $b_{\text{sin_log}} \in \mathbb{R}^{|I| \times |I| \times 1}$. Then, we add E^{Sin} to E^{Log} to obtain $E^{\text{Sin_Log}}$ using weight β :

$$s_{l s \ell} = \beta_s \otimes s_{s \ell} + (1 - \beta_s) \otimes l_{s \ell}, \ell \in \{1, 2, \dots, L\} \tag{6}$$

Second, the paper calculates the weight γ using a method in Eq. (7):

$$\gamma = \sigma([E^{\text{Sin_Log}}, E^{\text{Exp}}]W_{\text{sin_log_exp}} + b_{\text{sin_log_exp}}) \tag{7}$$

where $W_{\text{sin_log_exp}} \in \mathbb{R}^{2d \times 1}$, $b_{\text{sin_log_exp}} \in \mathbb{R}^{|I| \times |I| \times 1}$. Finally, the paper adds $E^{\text{Sin_Log}}$ to E^{Exp} to obtain the relative embeddings $E^{\text{Rel}} \in \mathbb{R}^{|I| \times |I| \times 1}$ by weight γ in Eq. (8):

$$r_{s \ell} = \gamma_s \otimes s_{l s \ell} + (1 - \gamma_s) \otimes e_{s \ell}, \ell \in \{1, 2, \dots, L\} \tag{8}$$

3.3 Local Learning Representation Module

Based on the SASRec model [9], the output vector x from the top self-attentive block is used as the local learning representation for the user's dynamic preferences in a sequence. Due to the importance of the absolute positions of a user-item interaction of a sequence, the paper adds the absolute position embedding matrix $E^{\text{Abs}} = \{a_1; a_2; \dots; a_L\} \in \mathbb{R}^{|I| \times d}$ to the embedding item matrix $E \in \mathbb{R}^{|I| \times d}$. In the end, an input matrix $X^{(0)} = \{x_1; x_2; \dots; x_L\} \in \mathbb{R}^{|I| \times d}$ is obtained. The specific process is formalized as shown in Eq. (9):

$$x_{\ell}^{(0)} = m_{s_{\ell}} + a_{\ell}, \ell \in \{1, 2, \dots, L\} \tag{9}$$

Afterward, the paper inputs the matrix $X^{(0)}$ into stacked self-attentive blocks (SAB) to get the output of the b^{th} block which is defined as shown in Eq. (10):

$$X^{(b)} = \text{SAB}^{(b)}(X^{(b-1)}) \quad b \in \{1, 2, \dots, B\} \tag{10}$$

Next is the omission of the normalization layer with residual connections, where each self-attentive block is considered as a self-attentive layer (\bullet), and then the results are fed into the feed forward layer, and values are calculated as shown in Eqs. (11–14):

$$\text{SAB}(X) = \text{FFL}(\text{SAL}(X)) \tag{11}$$

$$X' = \text{SAL}(X) = \text{softmax}\left(\partial \otimes \frac{QK^T}{\sqrt{d}} + (1 - \partial) \otimes \frac{QK^T}{\sqrt{d}} \otimes \frac{QE^{\text{RelT}}}{\sqrt{d}}\right) \Delta \cdot V \tag{12}$$

$$\text{FFL}(X') = \text{ReLU}(X'W_1 + 1^T b_1)W_2 + 1^T b_2 \tag{13}$$

$$\partial = \sigma\left(\left[\frac{QK^T}{\sqrt{d}}, \frac{QK^T}{\sqrt{d}} \otimes \frac{QE^{\text{RelT}}}{\sqrt{d}}\right]W_{\text{abs_rel}} + b_{\text{abs_rel}}\right) \tag{14}$$

Here, X is the input matrix containing absolute positional information, $Q = XW_Q, K = XW_K, V = XW_V$, in which W_Q, W_K and W_V present different weight matrixes. To consider the influence of each interaction pairs in the sequence, the paper uses relative embedding E^{Rel} . Moreover, the weight ∂ is introduced to balance the effect of the absolute embeddings and the relative embeddings. In Eq. (14), $W_{\text{abs_rel}} \in \mathbb{R}^{2d \times 1}$ and $b_{\text{abs_rel}} \in \mathbb{R}^{d \times 1}$, $[\cdot, \cdot]$ denotes the ternary concatenation operation, σ is the sigmoid function. Using the method, the paper can control the influence of the relative embeddings.

To improve flexibility, W_1, W_2 are the weights, and b_1, b_2 are biases of the two convolution layers. In the above equations, $1 \in \mathbb{R}^{d \times 1}$ is a unit vector, and Δ is a unit lower triangular matrix to be used as a mask, preserving only the transformation of the previous step. Meanwhile, the paper introduces a layer of normalization to normalize the output and dropout to avoid over-fitting, in which specific usage of layer normalization and dropout is the same as those in the SASRec [9].

3.4 Global Representation Learning Layer

Inspired by the paper [39], the paper proposes a novel attention module, called the location-based attention layer LBA (\bullet), to obtain the global learning representation of all users. In particular, the user u 's preference for interaction item s is generated in step $l + 1$ as a uniform aggregation of other interaction item representations, so the predicted ratings can be considered as the factorial similarity between user u 's historical item and candidate item s , as shown in Eq. (15):

$$\tilde{y}_{l+1}^u = \frac{1}{\sqrt{|S^u \setminus \{s_{l+1}^u\}|}} \sum_{i' \in S^u \setminus \{s_{l+1}^u\}} m_{i'} \tag{15}$$

In this paper, to calculate the most representative feature representation of all items in all sequences, the paper introduces $q^S \in \mathbb{R}^{1 \times d}$ as the query vector, rather than an average weighted aggregation. The q^S is different from Q in Eq. (12). This is because this module aims to calculate all users' global preferences instead of individual preferences.

The global representation of the sequences is formalized in Eq. (16):

$$y = \text{LBA}(E) = \text{softmax}\left(q^S (EW'_K)^T\right) EW'_V \tag{16}$$

where E is the initial input matrix and W'_K, W'_V are the learnable weight matrixes, similar to W_Q, W_K, W_V .

The dropout layer in the training process can avoid over-fitting and generalize the global representation in all steps, and then obtain the final global representation of all users' matrix Y , as shown in Eq. (17):

$$y_l = \text{Dropout}(y)l \in \{1, 2, \dots, L\} \quad (17)$$

3.5 Item Similarity Gating Layer

Designing the item similarity gating module has two main reasons. First, the global representation and the local representation are all important to the final recommendation. The paper must distinguish the importance between the global learning representation and the local learning representation. Second, the problem of uncertainty in users' intent in SR when user sees candidate items commonly exists.

Inspired by the NAIS [40], the paper designs a novel item similarity gating module which is critical in coordinating the local preference of the specific user and the global preference of all users. Meanwhile, it considers the influence of candidate items by calculating the similarity of candidate items and recently interacted items. The specific computing equation is formalized as follows in Eq. (18):

$$g = \sigma([m_{sl}, y, m_i]W_G + b_G) \quad (18)$$

where y represents the global representation, m_{sl} is the recently interacted item embedding vector, and m_i is the candidate item embedding vector. W_G and b_G are weight and bias separately. The paper utilizes the sigmoid function as the activation function.

At step l of the sequence, the final representation z_i is the weighted sum of the related local learning representation x_i and the global learning representation y , as shown in Eq. (19):

$$Z_l = x_i \otimes g + y \otimes (1 - g) \quad (19)$$

where \otimes is the broadcast operation in TensorFlow.

The final prediction of the item i preference to be the $(i + 1)^{\text{th}}$ item in the sequence is shown in Eq. (20):

$$R_{l+1,i} = z_l(m_i)^T \quad (20)$$

MTSR-FLP is trained using the Adam optimizer, which loss function is shown in Eq. (21):

$$L = - \sum_{u \in U} \sum_{l=1}^{L-1} \delta(s_{l+1}^u) \left[\log(\sigma(r_{l+1,s_{l+1}}^u)) + \log(1 - \sigma(r_{l+1,j}^u)) \right] \quad (21)$$

in which $r_{l+1,j}^u$ is the negative sample's possibility as the $(i + 1)^{\text{th}}$ item in the sequence, and $\delta(s_{l+1}^u)$ is used as eliminating the influence of padding items. If s_{l+1}^u is a padding item, $\delta(s_{l+1}^u)$ will be 0, otherwise, it will be 1.

Table 2 Statistics of the processed datasets

Dataset	#Users	#Items	#Interactions	Avg. length	Density (%)
Games	29,341	23,464	280,945	9.58	0.04
Beauty	40,226	54,542	353,962	8.8	0.02
Foursquare	22,748	11,146	145,106	6.38	0.06
Tmall	201,139	97,636	1,936,790	9.63	0.01
Steam	281,428	13,044	3,488,899	12.4	0.10

Table 3 Parameter descriptions and optimal settings

Parameter name	Parameter description	Optimal value
Num_epochs	Number of epochs	450
Batch_size	Batch size	128
Lr	Learning rate	0.001
Maxlen	Sequence length maximum	50
Hidden_units	Hidden vector dimension	50
Num_blocks	Number of self-attended blocks	2
Num_heads	Number of attention heads	1
Dropout_rate	dropout rate	0.5
l2_reg	L2 regularization	0

4 Experimental Results and Analysis

4.1 Experimental Environment

The related experiments environment of the proposed model is based on versions Python 3.6 and above and torch 1.10.0 or higher, and the running environment version requires Anaconda 3-2020.02 and above. The main data packages include cuda 10.2, cudnn10.2, torch==1.10.0+cu102, networkx==2.5.1, numpy==1.19.2, pandas==1.1.5, six==1.16.0, scikit-learn==0.24.2, and space==3.4.0. The hardware environment of this model is processor Intel Xeon Gold 5218, GPU is Rtx 3060 and memory is 32 GB DDR4.

This paper verifies the proposed model performance under strong generalization conditions by dividing all users into training/testing sets in the ratio of eight to two. In the user's historical sequence, the last item in the user's historical sequence is used in the testing set, and other items are used in the training set. During the process of training, the model is evaluated using the following two commonly used metrics. This paper conducts experiments on five real-world datasets—Beauty [41], Games [42], Steam [43], Foursquare [44] and Tmall [45]. The specific statistics are shown in Table 2. The optimal values of the nine parameters shown in Table 3 are the conclusions drawn through experimental comparison.

4.2 Evaluation Indicators

4.2.1 Click-Through Rate

This paper uses click-through rate (HR) to evaluate the percentage of recommended items that incorporate the correct item for at least one user interaction, and the metric is calculated as shown in Eq. (22):

$$HR@N = \frac{1}{|U|} \sum_{u \in U} \delta(|\hat{I}_{u,N} \cap T_u| > 0) \tag{22}$$

in which $\delta(\cdot)$ is the indicator function and equal to 1 if $|\hat{I}_{u,N} \cap T_u| > 0$.

4.2.2 Normalized Discounted Cumulative Gain

The paper takes the normalized discounted cumulative gain (NDCG@10) because it takes into account the location of correctly recommended items, and is calculated as shown in Eq. (23):

$$NDCG@N = \frac{1}{Z} DCG@N = \frac{1}{Z} \frac{1}{|U|} \sum_{u \in U} \sum_{k=1}^N \frac{\delta(\hat{i}_{u,k} \in I_u)}{\log_2(k+1)} \tag{23}$$

where $\hat{i}_{u,k}$ denotes the k^{th} item recommended to user u , and Z is a normalization constant indicating the discounted cumulative gain (NDCG@10) of the ideal case, which is the maximum possible value of NDCG@10.

4.3 Baseline Model

The following baseline models are utilized to be compared with the proposed MTSR-FLP model in this paper:

- BPRMF [46]: a standard recommendation model based on MF via pairwise loss.

- FISM [47]: another popular model learning the similarity matrix between items.
- GRU4Rec [1]: one of the earliest improved RNN models based on deep learning via an additional sampling strategy and twice loss function for sequence recommendation.
- SASRec [9]: a popular sequential recommendation model using a self-attention network.

4.4 Experimental Results

As shown in Table 4, the MTSR-FLP model achieves better results in both HR@10 and NDCG@10 compared with other sequence recommendation models. The performance comparison of MTSR-FLP and SASRec is shown in Figs. 3 and 4.

From Figs. 3 and 4, the MTSR-FLP’s performance in five datasets is much better than those of the SASRec. In the foursquare dataset, the performance SASRec is better than MTSR-FLP. This might be because that the results are influenced by over-fitting with the longer training time.

4.5 Ablation Study

As shown in Table 5, model-1 is the proposed model, the other models have different attention factors including absolute embeddings, relative embeddings, and calculating approaches. The performances of models with different settings are described in Table 6.

In the model-2, the paper defines E^{Pos} as the absolute embedding, E^{Exp} as the relative embedding. Since model-2 has only one absolute embedding and one relative embedding, the E^{Abs} is the same as E^{Pos} and the E^{Rel} is the same as E^{Exp} . In other words, the absolute embedding calculation block and the relative embedding calculation block are useless. Therefore, Eq. (12) is replaced by Eq. (24):

Table 4 HR@10 and NDCG@10 performance comparison

Dataset	Metric	BPRMF	FISM	GRU4Rec	SASRec	MTSR-FLP	MTSR-FLP vs. SASRec
Tmall	HR@10	0.1744	0.2149	0.3526	0.3844	0.4382	14.00%
	NDCG@10	0.0825	0.1043	0.2072	0.2123	0.2462	15.97%
Steam	HR@10	0.1023	0.3183	0.3177	0.3424	0.4489	31.10%
	NDCG@10	0.0468	0.1703	0.1707	0.1789	0.255	42.54%
Games	HR@10	0.3454	0.4791	0.4825	0.5769	0.6662	15.49%
	NDCG@10	0.1981	0.2631	0.2906	0.3507	0.4134	17.88%
Beauty	HR@10	0.2498	0.3533	0.2729	0.3433	0.4311	25.58%
	NDCG@10	0.1148	0.1942	0.1683	0.2119	0.2658	25.41%
Foursquare	HR@10	0.2659	0.3977	0.4324	0.4767	0.5095	6.88%
	NDCG@10	0.1287	0.2025	0.2375	0.2611	0.2818	8.60%

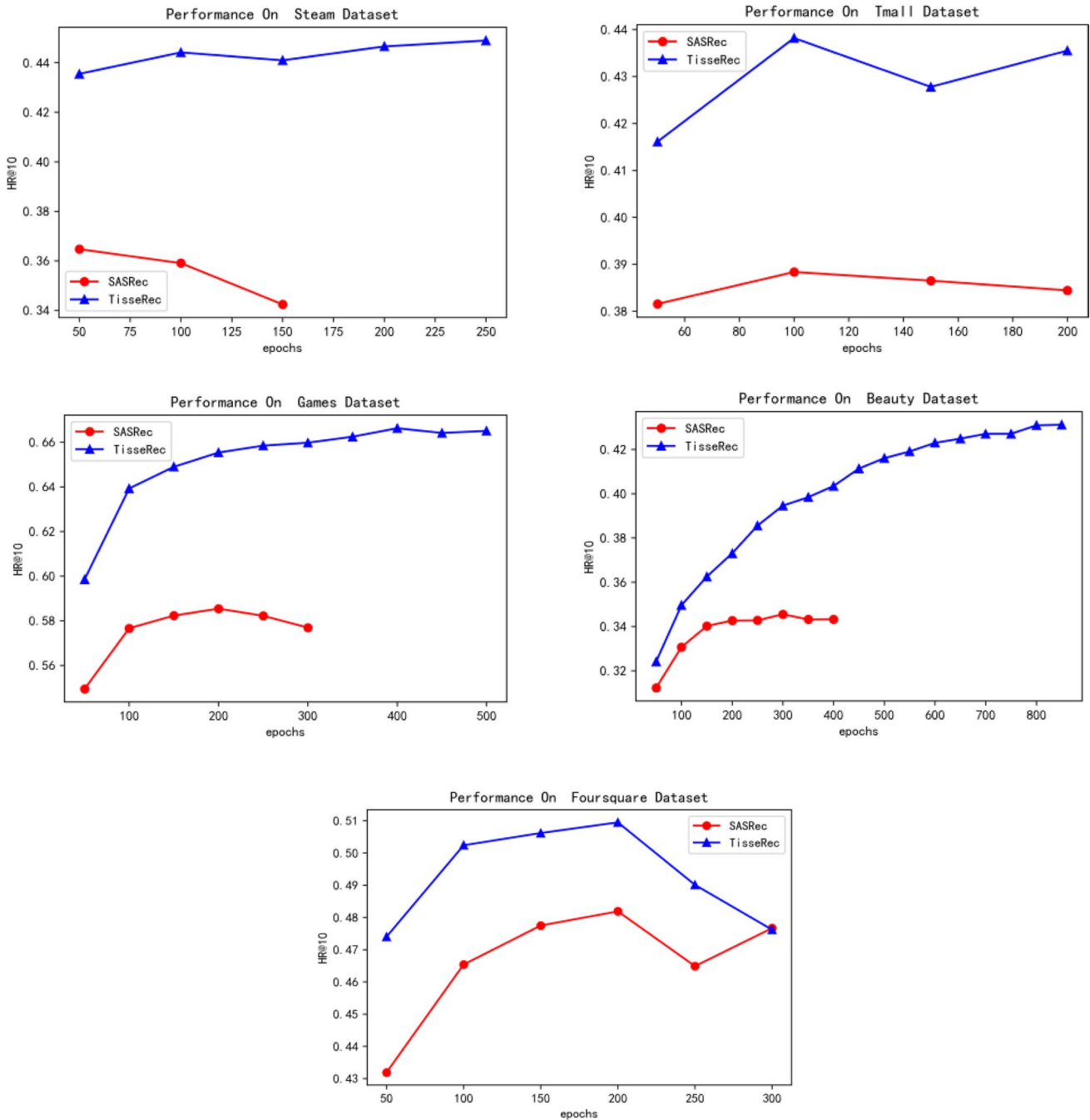


Fig. 3 SASRec model and MTSR-FLP model on five different datasets performance on HR@10. The blue dotted line represents MTSR-FLP and the red dotted line represents the SASRec model. It can be

seen that under the evaluation index of NDCG@10, MTSR-FLP performs better than the SASRec model in the five datasets

$$X^l = SAL(X) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}} \otimes \frac{QE^{RelT}}{\sqrt{d}}\right) \Delta \cdot V \quad (24)$$

Compared with Eq. (12), there is no parameter ∂ which means model-2 does not consider the negative influence

of relative embeddings. Model-3 is similar to the model-2 except that E^{Exp} is replaced with E^{Sin} .

In model-4, the paper uses E^{Pos} as absolute embedding and uses E^{Exp} and E^{Log} as the relative embeddings, in which E^{Abs} is same as E^{Pos} , so the absolute embedding calculation

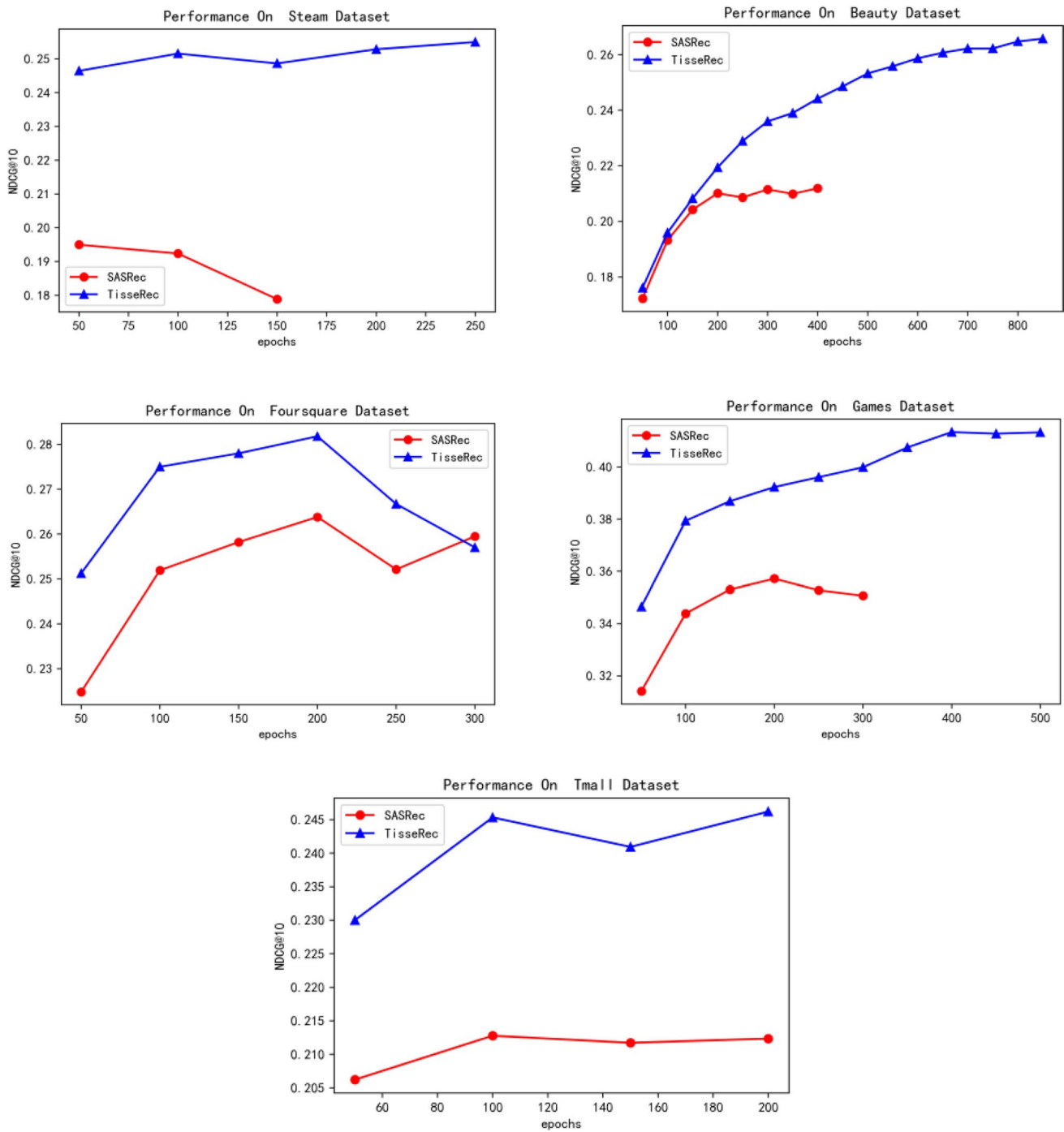


Fig. 4 SASRec model and MTSR-FLP model on five different datasets performance on NDCG@10. It can be seen that under the evaluation index of NDCG@10, MTSR-FLP performs better than the SASRec model in the five datasets

Table 5 MTSR-FLP models with different settings

Model ID	Absolute embeddings	Relative embeddings	Calculating approaches
Model-1	E^{Day}, E^{Pos}	$E^{Log}, E^{Sin}, E^{Exp}$	Product, add by weight
Model-2	E^{Pos}	E^{Exp}	Product
Model-3	E^{Pos}	E^{Sin}	Product
Model- 4	E^{Pos}	E^{Exp}, E^{Log}	Product, forward neural network

Table 6 Performance of models with different settings

Dataset	Metric	Model-1	Model-2	Model-3	Model-4
Tmall	HR@10	0.4382	0.4646	0.3974	0.4237
	NDCG@10	0.2462	0.263	0.2178	0.2338
Steam	HR@10	0.4489	0.4276	0.4412	0.4328
	NDCG@10	0.255	0.2406	0.2492	0.2474
Games	HR@10	0.6662	0.6689	0.6611	0.6577
	NDCG@10	0.4134	0.4078	0.4015	0.396
Beauty	HR@10	0.4311	0.4174	0.3965	0.3964
	NDCG@10	0.2658	0.254	0.2335	0.2335
Foursquare	HR@10	0.5095	0.4576	0.4773	0.4806
	NDCG@10	0.2818	0.2447	0.2561	0.258

block is useless. The relative embedding calculation block is different from that of the proposed model. First, the paper concatenates the E^{Sin} and E^{Log} together to get E^{Rel} . Then, the paper feeds E^{Rel} into the feed forward neural network to get

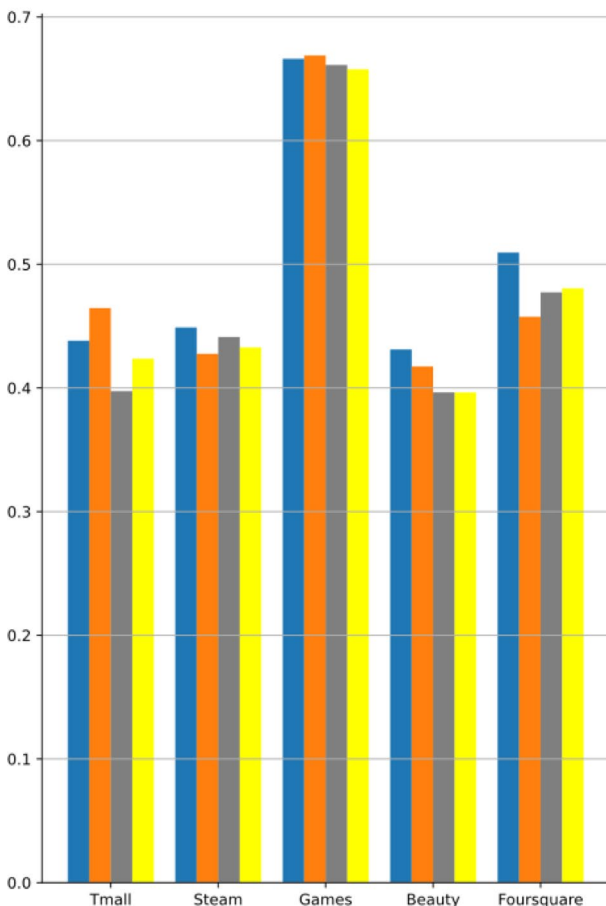
the final E^{Rel} . Model-4 does not consider the negative influence of the relative embeddings. This can be formalized by following Eq. (25):

$$E^{Rel} = FFL([E^{Sin}, E^{Log}]) \quad (25)$$

The performance of different settings of the model is shown in Table 6. To highlight the training performance, the paper depicts the results in Fig. 5. From the Fig. 5 the proposed model performs best in almost all datasets on HR@10 and NDCG@10. The specific training process is shown below. The paper can see the proposed model's performance is better than the model with other settings in almost all datasets.

The specific training process is shown in Figs. 6 and 7. It is easy to see that the proposed model's performance is better than the model with other settings in almost all datasets. The analysis of ablation experimental results is described as follows.

HR@10



NDCG@10

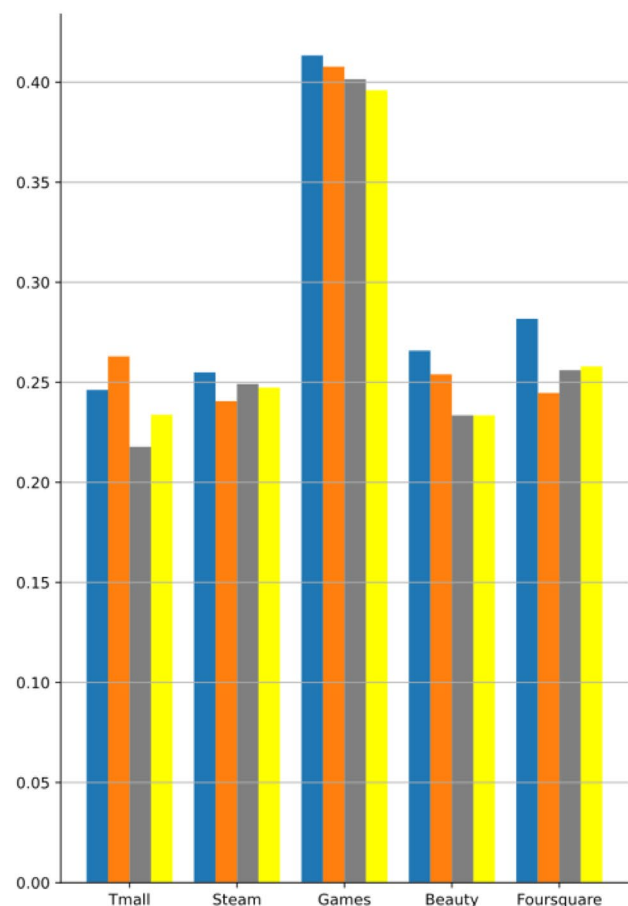


Fig. 5 The performance of MTSR-FLP with different settings. In both HR@10 and NDCG@10, Model-1 performed very well. Except for Tmall datasets, Model-1 outperforms other models on the other four datasets

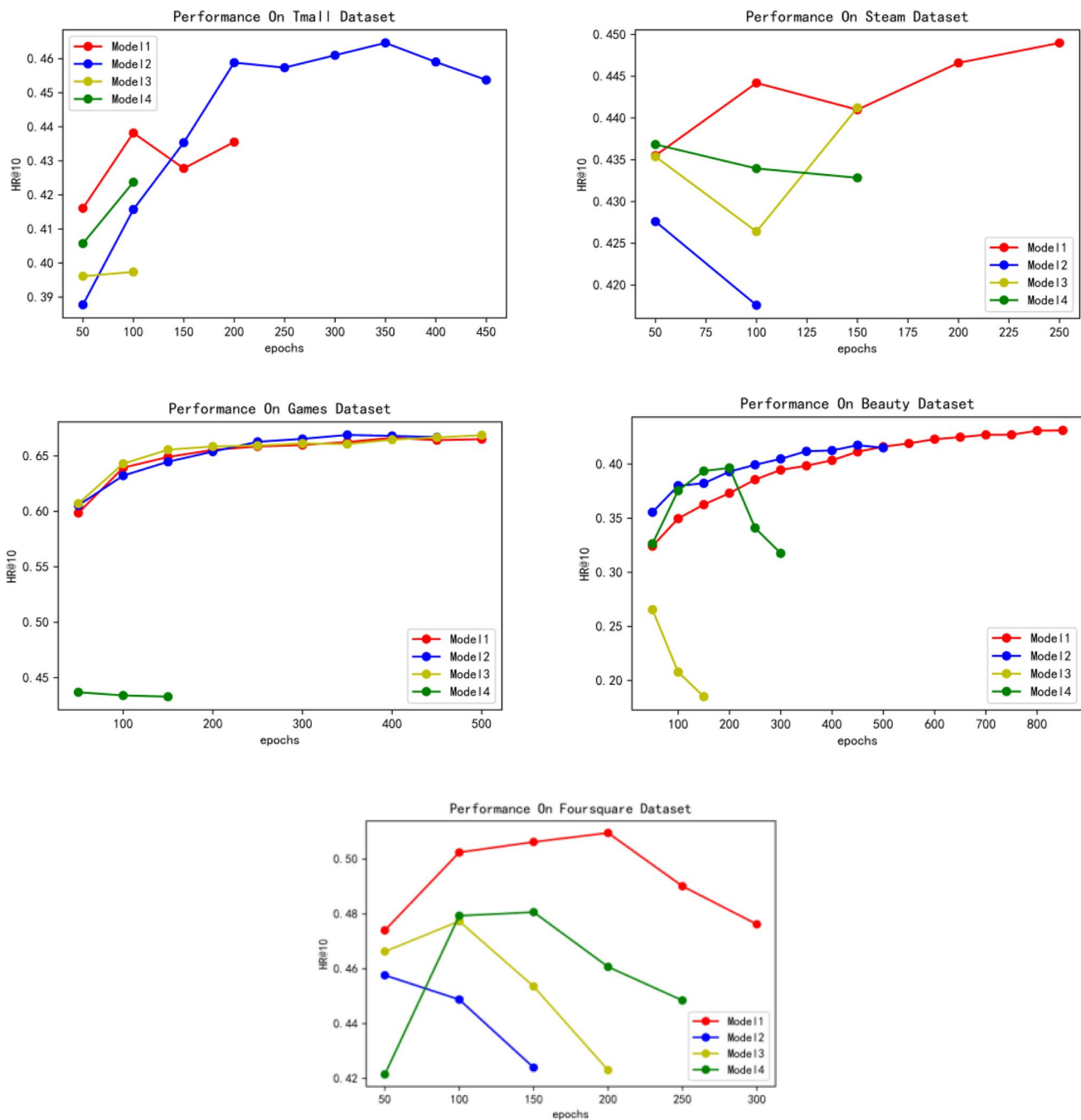


Fig. 6 The performance in different HR@10 for MTSR-FLP models with different settings. Model-2 performed well on the first datasets. Model-1 outperformed the other models on the remaining four datasets

- Model-1 vs. Model-2 and Model-3. In the training process, the paper can see the performance of Model-1 is better than Model-2 and Model-3 in most datasets. The reason for this is that Model-1 only uses E^{Exp} as relative embedding and the Model-2 only use E^{Log} as relative embedding, but the proposed model use E^{Exp} , E^{Log} , and E^{Sin} as relative embeddings. Therefore, the proposed model uses more information than Model-2 and Model-3.
- Model-2 vs Model-3. In different datasets, the performance of Model-2 and Model-3 is different. This is mainly because the E^{Exp} and E^{Sin} capture different time information. If the paper uses E^{Exp} as relative embedding, a large time gap can quickly decay to zero. But if

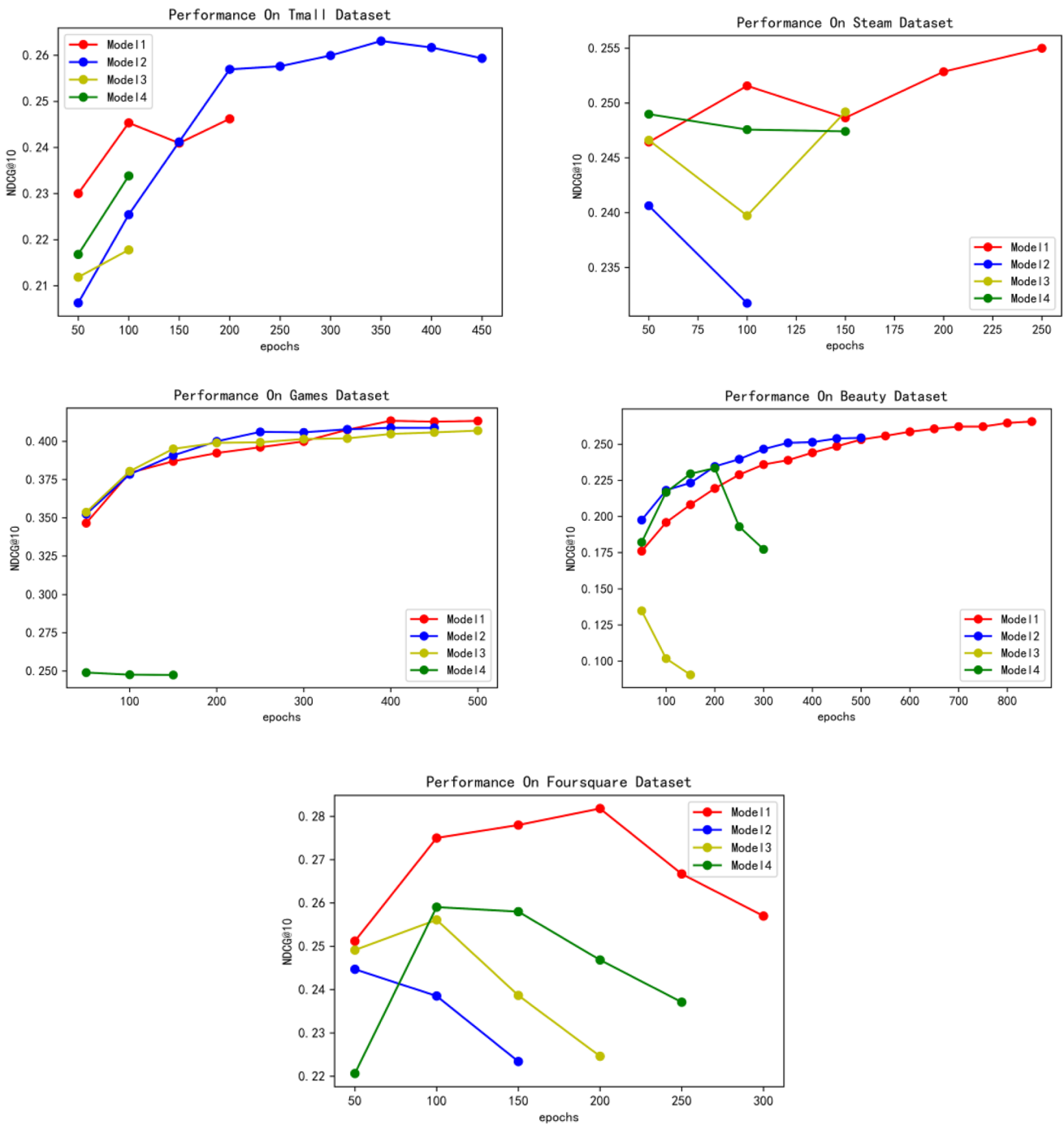


Fig. 7 The performance in different epochs of NDCG@10 for MTSR-FLP models with different settings

the paper uses E^{Sin} as relative embedding, the relative embedding only captures periodic occurrences. Time information is different in different datasets, so the performance in different datasets is different.

- Model-4 vs Model-2 and Model-3. Compared with Model-2 and Model-3, Model-4 performs worse than Model-2 and Model-3. Though Model-4 uses more infor-

mation than Model-2 and Model-3, combining E^{Exp} and E^{Log} using feed forward network causes over-fitting.

- Model-1 vs Model-2, Model-3 and Model-4. From Table 6, it can be seen that the performance of Model-2, Model-3, and Model-4 in datasets foursquare is worse than the proposed model. In these models, they do not consider the negative influence of relative

embedding. But in this datasets, relative time information is useless.

5 Conclusion

This paper proposes a novel MTSR-FLP model for the sequence recommendation. In particular, the paper integrates attention mechanisms, including multi-temporal embeddings self-attention network, and a factorial item similarity, to improve the performance of the SR. In particular, the multiple temporal embedding provides the absolute embeddings and relative embeddings for the self-attention network using temporal information and multiple encoding functions, where different attention heads handle different temporal embeddings. Meanwhile, this paper also designs a gating module for modulating local and global representations, considering the relationship between candidate items, recent interacted items, and all user's global preferences to cope with possible uncertainty in user intent.

In addition, there are some directions to improve the proposed model. First, although this paper has tried deep learning and self-attention network models, more strategies might be worth trying to improve the performance of SR. In the near future, the paper will try integrating some new attention mechanisms, such as GNNs, and generative adversarial networks (GANs) into the MTSR-FLP model. Second, future works should capture the multi-dimension information of items and the users in the SR, which can be utilized to further classify recommendation items and the users. In addition, more detailed time feature information also needs to be obtained to improve the accuracy of the SR. Finally, the paper will utilize new datasets to evaluate and improve the performance of the proposed model.

Acknowledgements The authors would like to thank the anonymous reviewers for their comments which helped us to improve the quality of the paper.

Author Contributions All the authors have contributed to the design and evaluation of the schemes and the writing of the manuscript. All the authors have read and approved the final manuscript.

Funding This paper is supported by Key scientific research projects of colleges and universities in Henan Province (Grand No. 23A520054).

Data Availability Data sharing is not applicable to this article as no real-world datasets were acquired or analyzed during the current study.

Declarations

Conflict of Interest The authors of the paper certify that they have no affiliations with or involvement in any organization or entity with any financial interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D., Japa.: Session-based recommendations with recurrent neural networks (2015) Doi: <https://doi.org/10.48550/arXiv.1511.06939>
- Le, DT., Lauw, HW., Fang, Y.: Modeling contemporaneous basket sequences with twin networks for next-item recommendation. (2018) Doi: <https://doi.org/10.24963/ijcai.2018/474>
- Chen, X., Xu, H., Zhang, Y., et al.: Sequential Recommendation with User Memory Networks. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (2018) Doi: <https://doi.org/10.1145/3159652.3159668>
- Jannach, D., Ludewig, M.: When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation. In: Proceedings of the Eleventh ACM Conference on Recommender Systems; 2017. Doi: <https://doi.org/10.1145/3109859.3109872>
- He, R., McAuley, J.: Fusing similarity models with markov chains for sparse sequential recommendation. In: Paper presented at: 2016 IEEE 16th International Conference on Data Mining (ICDM)2016. Doi: <https://doi.org/10.1109/ICDM.2016.0030>
- Zhang, S., Yao, L., Sun, A., Tay, Y.: Deep learning based recommender system. ACM Comput. Surv. **52**(1), 1–38 (2019). <https://doi.org/10.1145/3285029>
- He, X., He, Z., Du, X., Chua, T-S.: Adversarial Personalized Ranking for Recommendation. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (2018) Doi: <https://doi.org/10.1145/3209978.3209981>
- Wang, S., Cao, L., Wang, Y., Sheng, Q.Z., Orgun, M.A., Lian, D.: A survey on session-based recommender systems. ACM Comput. Surv. **54**(7), 1–38 (2022). <https://doi.org/10.1145/3465401>
- Kang, W-C., McAuley, J.: Self-Attentive Sequential Recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM) (2018) Doi: <https://doi.org/10.1109/ICDM.2018.00035>
- Feng, F., He, X., Wang, X., Luo, C., Liu, Y., Chua, T-SJAToIS.: Temporal relational ranking for stock prediction (2019) 37(2): 1–30. Doi: <https://doi.org/10.1145/3309547>
- Wu, C.-Y., Ahmed, A., Beutel, A., Smola, AJ., Jing, H.: Recurrent recommender networks. In: Paper presented at: Proceedings of the tenth ACM international conference on web search and data mining (2017) Doi: <https://doi.org/10.1145/3018661.3018689>
- Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. In ICLR (2016). <https://doi.org/10.48550/arXiv.1511.06939>
- Hidasi, B., Karatzoglou, A.: Recurrent neural networks with top-k gains for session-based recommendations. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM) (2018) p. 843–852. Doi: <https://doi.org/10.1145/3269206.3271761>
- Yuan, F., Karatzoglou, A., Arapakis, I., Jose, J.M., He, X.: A simple convolutional generative network for next item

- recommendation. Paper Present (2019). <https://doi.org/10.1145/3289600.3290975>
15. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. Paper presented at: Advances in neural information processing systems (2017) Doi: <https://doi.org/10.48550/arXiv.1706.03762>
 16. Devlin, J., Chang, M. W., Lee, K., et al.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018) Doi: <https://doi.org/10.48550/arXiv.1810.04805>
 17. Fei, S., Jun, L., Jian, W., Changhua, P., Xiao, L., Wenwu, O., Peng, J.: BERT4Rec: sequential recommendation with bidirectional encoder representations from transformer. CIKM. (2019). <https://doi.org/10.1145/3357384.3357895>
 18. Chengfeng, Xu., Feng, J., Zhao, P., Zhuang, F., Wang, D., Liu, Y., Sheng, V.S.: Long and short-term self-attention network for sequential recommendation. *Neurocomputing* **423**(2021), 580–589 (2021). <https://doi.org/10.1016/j.neucom.2020.10.066>
 19. Xiang, W., Xiangnan, H., Meng, W., Fuli F., Tat-Seng, C.: Neural Graph Collaborative Filtering. In SIGIR. (2019) 165–174. Doi: <https://doi.org/10.1145/3331184.3331267>
 20. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: simplifying and powering graph convolution network for recommendation. ACM (2020). <https://doi.org/10.1145/3397271.3401063>
 21. Shu, W., Yuyuan, T., Yanqiao, Z., Liang, W., Xing, X., Tieniu, T.: Session-Based Recommendation with Graph Neural Networks. In AAAI. 8 pages (2019) Doi: <https://doi.org/10.1609/aaai.v33i01.3301346>
 22. Wang, Z., Wei, W., Cong, G., Li, X.-L., Mao, X.-L., Qiu, M.: Global Context Enhanced Graph Neural Networks for Session-based Recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, (2020) pp. 169–178. Doi: <https://doi.org/10.1145/3397271.3401142>
 23. Rianne van den Berg, Thomas N. Kipf, and Max Welling.: Graph convolutional matrix completion. arXiv:1706.02263. Retrieved from <https://arxiv.org/abs/1706.02263> (2017) Doi: <https://doi.org/10.48550/arXiv.1706.02263>
 24. Nguyen G. H., Lee J. B., Rossi R. A., Ahmed N. K., Koh E., Kim S.: “Continuous-Time Dynamic Network Embeddings” In: 12 Web Conference 2018—Companion of the World Wide Web Conference, WWW 2018, no. BigNet, pp. 969–976, 2018. Doi: <https://doi.org/10.1145/3184558.3191526>
 25. Lo, Y.-Y., Liao, W., Chang, C.-S., Lee, Y.-CJIToCSS.: Temporal matrix factorization for tracking concept drift in individual user preferences (2017) 5(1): 156–168. Doi: <https://doi.org/10.1109/TCSS.2017.2772295>
 26. Koren, Y.: Collaborative filtering with temporal dynamics. In: Paper presented at: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining2009. Doi: <https://doi.org/10.1145/1557019.1557072>
 27. Dallmann, A., Grimm, A., Pölitz, C., Zoller, D., Hotho, AJapa.: Improving session recommendation with recurrent neural networks by exploiting dwell time (2017) Doi: <https://doi.org/10.48550/arXiv.1706.10231>
 28. Zhu, Y., Li, H., Liao, Y., et al.: What to Do Next: Modeling User Behaviors by Time-LSTM. In: Paper presented at: IJCAI2017 Doi: <https://doi.org/10.24963/ijcai.2017/504>
 29. Zhang, S., Tay, Y., Yao, L. et al.: Next item recommendation with self-attentive metric learning. Thirty-Third AAAI Conference on Artificial Intelligence (2019), Doi: <https://doi.org/10.48550/arXiv.1808.06414>
 30. Wu, J., Cai, R., Wang, H. Déjà, vu: A contextualized temporal attention mechanism for sequential recommendation. In: Paper presented at: Proceedings of The Web Conference 2020. Doi: <https://doi.org/10.1145/3366423.3380285>
 31. Zhou, C., Bai, J., Song, J., et al.: Atrank: An attention-based user behavior modeling framework for recommendation. In: Paper presented at: Proceedings of the AAAI Conference on Artificial Intelligence2018. Doi: <https://doi.org/10.1609/aaai.v32i1.11618>
 32. Li, J., Wang, Y., & McAuley, J.: Time interval aware self-attention for sequential recommendation. In: Proceedings of the 13th international conference on web search and data mining (pp. 322–330). (2020). Doi: <https://doi.org/10.1145/3336191.3371786>
 33. Ye, W., Wang, S., Chen, X., Wang, X., Qin, Z., & Yin, D. (2020). Time matters: Sequential recommendation with complex temporal information. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval (pp. 1459–1468). Doi: <https://doi.org/10.1145/3397271.3401154>
 34. Wang, C., Ma, W., Zhang, M., Chen, C., Liu, Y., Ma, S.: Toward dynamic user intention: Temporal evolutionary effects of item relations in sequential recommendation ACM Transactions on Information Systems (TOIS), 39 (2020), pp. 1–33. Doi: <https://doi.org/10.1145/3432244>
 35. Hsu, C., Li, C.-T.: Retaggn: Relational temporal attentive graph neural networks for holistic sequential recommendation. In: Proceedings of the web conference (2021) (pp. 2968–2979). Doi: <https://doi.org/10.1145/3442381.3449957>
 36. Wang, H., Li, P., Liu, Y., Shao, J.: Towards real-time demand-aware sequential poi recommendation. *Inf. Sci.* **547**, 482–497 (2021). <https://doi.org/10.1016/j.ins.2020.08.088>
 37. Chen, Z., Zhang, W., Yan, J., Wang, G., & Wang, J. (2021). Learning dual dynamic representations on time-sliced user-item interaction graphs for sequential recommendation. In: Proceedings of the 30th ACM international conference on information & knowledge management (p. 231–240). Doi: <https://doi.org/10.1145/3459637.3482443>
 38. Jin, J., Chen, X., Zhang, W., Huang, J., Feng, Z., & Yu, Y. (2022). Learn over past, evolve for future: Search-based time-aware recommendation with sequential behavior data. In: Proceedings of the ACM web conference 2022 (pp. 2451–2461). Doi: <https://doi.org/10.1145/3485447.3512117>
 39. Cho, S.M., Park, E., Yoo, S.: MEANTIME: mixture of attention mechanisms with multi-temporal embeddings for sequential recommendation. Paper Present (2020). <https://doi.org/10.1145/3383313.3412216>
 40. Lin, J., Pan, W., Ming Z. FISSA: Fusing Item Similarity Models with Self-Attention Networks for Sequential Recommendation. Fourteenth ACM Conference on Recommender Systems; 2020. Doi: <https://doi.org/10.1145/3383313.3412247>
 41. He, X., He, Z., Song, J., et al.: Nais: Neural attentive item similarity model for recommendation. (2018) 30(12): 2354–2366. Doi: <https://doi.org/10.1109/TKDE.2018.2831682>
 42. Beauty. <http://jmcauley.ucsd.edu/data/amazon/>. Accessed 12 July 2023
 43. Games. <http://jmcauley.ucsd.edu/data/amazon/>. Accessed 12 July 2023
 44. Steam. https://cseweb.ucsd.edu/~jmcauley/datasets.html#steam_data. Accessed 12 July 2023
 45. Foursquare. https://archive.org/download/201309_foursquare_dataset_umn. Accessed 12 July 2023
 46. Tmall. <https://tianchi.aliyun.com/dataset/>. <https://doi.org/10.48550/arXiv.1205.2618>. Accessed 12 July 2023
 47. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.J. BPR: Bayesian personalized ranking from implicit feedback (2012) Doi: <https://doi.org/10.48550/arXiv.1205.2618>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.