Knowledge-Based Systems 230 (2021) 107375

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

GFE: General Knowledge Enhanced Framework for Explainable Sequential Recommendation

Zuoxi Yang, Shoubin Dong [*], Jinlong Hu

Guangdong Key Lab of Computer Network, School of Computer Science & Engineering, South China University of Technology, Guangzhou, China

ARTICLE INFO

Article history:
Received 30 March 2021
Received in revised form 5 July 2021
Accepted 9 August 2021
Available online 12 August 2021

Keywords:
Explainable recommendation
Hierarchical self-attention
Knowledge graph
Sequential recommendation

ABSTRACT

It is vital for sequential recommendation to provide accurate and explainable results for user, which can help them make better decisions. In this paper, we develop a General Knowledge Enhanced Framework for Explainable Sequential Recommendation (GFE) to capture user's fine-grained preferences and dynamic preferences evolution. Specifically, the fine-grained preferences are modeled as intrinsic interests and external potential interests, which can be captured by sequential-aware interest and knowledge-aware interest modules respectively. Moreover, the high-order paths between each user-item pair are generated with the help of the knowledge graph, which contain abundant high-order semantic relevance among entities. To make better use of this character, we propose a hierarchical self-attention mechanism to aggregate the high-order semantic information from these knowledge paths, thus discovering user's dynamic preferences evolution. According to these abilities, GFE can provide more reasonable explanations from the views of microcosm and macrocosm. Unlike other traditional explainable sequential recommendation models, GFE has the strong generalization to integrate with other pure sequential recommendation models and endow explainability to them. Based on this nature, we combine the GFE with two state-of-the-art sequential recommendation models and further propose two GFE-based models, called GFE-SASRec and GFE-TiSASRec, to show the availability of GFE. Finally, experiments conducted on three real-world public datasets demonstrate the state-of-the-art performance and the strong explainability of GFE.

1. Introduction

Sequential recommendation has been adopted to meet user's preferences through his/her chronological historical behaviors [1]. Accuracy and explainability are two important factors of sequential recommendation. To improve the accuracy of the sequential recommendation, it is crucial to capture user's fine-grained preferences and dynamic preferences evolution since their preferences are not unchangeable in practice. Moreover, it is important to note as well that offering reasonable explanations can help user to understand the reasons behind the recommend system recommends particular items to them, thus facilitating them to do better decisions.

Among the previous literature, the Markov Chain is used to model user's sequential behaviors [2,3], which assumes that the user's next behavior depends on his/her previous sequences. For example, FPMC [1] is able to take the advantages of the Markov Chain and matrix factorization (MF) to capture user's long-term preferences and short-term item transitions. Although most of them can perform well in high-sparsity settings, they may be not suitable for complex dynamic scenarios. Neural networks also have been proven their strong ability in sequential recommendation. Some researchers [4,5] adopt convolutional neural networks (CNN) to model user's previous items as an "image" and extract their sequential patterns by using convolutional filters. Another line to model user's sequential behaviors is recurrent neural networks (RNN), which can improve the recommendation performance by encoding user's behaviors as hidden states, such as [6–11]. Recently, transformer has achieved great success in the natural language processing (NLP) field [12], which is purely based on self-attention mechanism. Inspired by it, several self-attention based models [13,14] have been proposed for the sequential recommendation. Both of them have shown state-of-the-art performance without any convolutional and recurrent modules. However, most of these approaches cannot capture the user's fine-grained preferences well.

Knowledge graph (KG) has drawn more and more attention in recent years due to its abundant real-world facts and entity connections, which can improve the accuracy and explainability of recommendation. One of the effective methods to incorporate the KG into recommendation is knowledge graph embedding (KGE) based models. They are capable to embed the entities and

[*] Corresponding author.
E-mail addresses: csyangzuoxi@mail.scut.edu.cn (Z. Yang), sbdong@scut.edu.cn (S. Dong).

https://doi.org/10.1016/j.knosys.2021.107375
0950-7051/© 2021 Elsevier B.V. All rights reserved.

relations of KG into low-dimensional continuous vector spaces via KGE techniques, such as TransE [15], TransH [16], TransR [17] and TransD [18]. KGE-based models [19–21] can generate a better representation for the item by leveraging its content representation and knowledge embeddings, further improving the recommendation performance. However, these methods have some limitations: (1) the hidden representations cannot provide explicit explanations for the target user that why the recommender system recommends such specific items to them. (2) most of them are more suitable for the general recommendation because they do not consider the sequential patterns, resulting in a lack of ability to capture user's preferences drifting. (3) the high-order semantic information among item entities is also ignored since the KGE-based methods focus more on the 1-order neighbor structural information. Besides, some researchers extend the knowledge graph to the explainable recommendation [22–24]. Although most of them are capable to offer explanations for the target user, their drawbacks are that: (1) they are proposed for the general recommendation without modeling the sequential behaviors. (2) such explainable methods lack good generalization since it is difficult for them to integrate with other pure sequential recommendation models.

To tackle these challenges of the above-mentioned work, we propose a General Knowledge Enhanced Framework for Explainable Sequential Recommendation (GFE). Firstly, GFE aims at exploring user's fine-grained preferences by modeling them as intrinsic interests and external potential interests. Specifically, there are two modules in GFE, called sequential-aware interest module and knowledge-aware interest module, which are designed to characterize these two different preferences respectively. Second, since the user's preferences are dynamic and changeable, sequential recommendation needs to capture their dynamic preference evolution. To achieve this goal, we generate knowledge-aware paths for each user-item pair with the help of the KG. Different paths between the specific user-item pair would contain different semantic interaction signs in different sequence contexts. Through these high-order paths, it is helpful for us to explore user's preferences in different time steps. Therefore, we can discover the user's dynamic preference evolution and further endow reasoning ability for the explainable sequential recommendation. After obtaining the user's fine-grained preferences and dynamic preferences evolution, the sequential recommendation can correspondingly offer strong explainable results from the two views of microcosm and macrocosm. Finally, it is worth pointing out that the GFE has strong generalization, which can integrate another pure sequential recommendation model in the sequential-aware interest module. Base on this, it is helpful to endow explainability for such pure sequential recommendation models even if they cannot provide explainable recommendation results originally.

To summarize, the contributions of this paper are as follows:

- To provide accuracy and explainable recommendation results, we propose a **General** Knowledge Enhanced **Framework** for **Explainable** Sequential Recommendation (GFE). Specifically, GFE aims at capturing user's fine-grained preferences and their user's dynamic preferences with the help of the KG.
- In the GFE, user's fine-grained preferences are further modeled as intrinsic interests and external potential interests, which are explored by the sequential-aware interest module and knowledge-aware interest module respectively. Moreover, in the knowledge-aware interest module, we propose a hierarchical self-attention mechanism to characterize the high-order semantic relevance among entities, thus discovering user's dynamic preferences evolution.
- GFE emphasizes the importance of the general framework for explainable sequential recommendation. Unlike other traditional explainable recommendation, GFE has strong generalization that can integrate with other pure sequential recommendation models and endow explainability to them. To verify its availability, we also propose two GFE-based models, called GFE-SASRec and GFE-TiSASRec, through combining the GFE to two pure state-of-the-art sequential models, i.e., SASRec and TiSASRec.
- Experiments conducted on three real-world datasets have demonstrated the state-of-the-art performances and the robustness of explainability of GFE.

2. Related work

In this section, we will introduce several previous studies that related to our model.

Unlike general recommendation [25–28], sequential recommendation [29–31] adopts user's sequential behaviors to predict user's next action. In the previous work, some researchers utilized Markov Chain to capture user's sequential patterns. For example, Rendle et al. [1] proposed Factorized Personalized Markov Chain (FPMC) to combine the Markov Chain with matrix factorization. It is capable to capture user's long-term interests and short-term transitions through the matrix factorization and a transition matrix. However, the transition matrix used in FPMC focused more on 1-order Markov-Chain. To extend this idea, He et al. [32] proposed Fossil model to use high-order Markov-Chain, which has shown stronger performance on sparse datasets. These Markov-Chain based methods may perform poorly to capture complicated dynamics in complex scenarios. With the success of neural networks, many deep learning based methods are leveraged for the sequential recommendation. Tang et al. [4] proposed Caser model, which adopted CNN to model user's recent behaviors as an "image" and learn the sequential patterns from these behaviors via using convolutional filters. However, Caser is a 1D CNN based approach. To extend this idea, Yan et al. [5] built a 2D CNN based model, CosRec, for the sequential recommendation to extract sequential features. It can encode sequential items' embeddings into pairwise representations to achieve better performance. Besides, RNN also has been widely utilized for the sequential recommendation. Hidasi et al. [7] proposed GRU4Rec model to encode user's item sequence into hidden states by using Gate Recurrent Units (GRU). In addition, Hidasi et al. [6] introduced a new loss function and an improved sampling strategy to improve the GRU4Rec's performance. To alleviate the short memory problem of the RNN, Li et al. [8] introduced the attention mechanism to capture user's general states and current states. Although these methods are capable to effectively encode user's behaviors into hidden vectors for the improvement of the sequential recommendation, most of them suffer from some problems, such as large time overhead, hard to parallelize and so on. Recently, Transformer has shown state-of-the-art performance and efficiency for the NLP task [12], which is purely based on the self-attention mechanism. Inspired by it, some researchers applied self-attention to the sequential recommendation. Kang et al. [13] proposed SASRec to model the entire user's sequential behaviors with self-attention, which significantly outperforms state-of-the-art Markov Chain/CNN/RNN based sequential recommendation methods without any convolutional or recurrent operations. However, SASRec did not consider the time intervals between each interaction. To extend this idea, Li et al. [14] proposed a time interval aware self-attention model, termed as TiSASRec. TiSASRec aimed at modeling both the items' absolute positions and their time intervals. However, most of them lack the strong ability to capture user's fine-grained preferences.

Knowledge graph has been drawn more and more attention in recent years, which can improve accuracy and explainability for the recommendation. To integrate knowledge graph into recommendation, many researchers adopted KGE-based techniques to encode entities and relations to representation vectors. For example, Zhang et al. [19] proposed CKE model to learn items' representations from their structural content, textual content and visual content by the TransR techniques. Wang et al. [20] proposed a news recommendation model, DKN, which utilized the TransD techniques to learn knowledge embeddings for enriching the news' representations. Wang et al. [33] proposed a multi-task feature learning model with the knowledge-enhanced, termed as MKR, which adopted an improved knowledge graph embedding module to improve the recommendation performance. Tang et al. [34] applied TransR technique to obtain proper representations for items and applied them to promote the recommendation performance. However, (1) these methods are more suitable for the general top-n recommendation but not the sequential recommendation, and they are hard to capture user's preferences drifting. (2) the KGE based models pay more attention to the 1-order neighbor structural information, which ignores the high-order semantic information among entities. They lack the strong ability to capture user's dynamic preferences evolution from the high-order paths. (3) it is difficult for them to provide intuitive explanations since the knowledge embeddings are implicit. Some researchers extended the knowledge graph to the explainable recommendation [22–24]. Although they could offer explainable results for users, they are also proposed for the general recommendation without considering the sequential patterns. Besides, they also lack good generalization since it is difficult for them to integrate with other pure sequential recommendation models. Some knowledge-enhanced models [35,36] are proposed for explainable sequential recommendation. But they are also limited by the good generalization to integrate with other pure recommendation models.

3. Proposed framework

In this section, we will introduce the GFE in detail, a general knowledge enhanced framework for explainable sequential recommendation. The framework is shown in Fig. 1, including three main components: (1) sequential-aware interest module: which aims at capturing user's intrinsic interests from his/her sequential behaviors (2) knowledge-aware interest module: which attends to capture user's external potential interests and their dynamic sequential evolution. (3) prediction module: which integrates user's sequential-aware interests and knowledge-aware interests to offer predicted results.

Generally, user's interactions are defined as a matrix $\mathrm{Y} = \{y_{ui} | u \in U, i \in I\}$. where U and I are the user set and item set respectively. In the sequential recommendation, we focus more on predicting whether the user will click the recommended items, so we apply implicit feedback for our model, which can be defined as follows:

$$y_{ui} = \begin{cases} 1, & \text{if the user } u \text{ clicked the item } i \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $y_{ui} = 1$ means that there is an observation between the user u and item i, i.e., positive feedback, and $y_{ui} = 0$ is viewed as a negative interaction.

For each user u, his/her sequential behaviors can be denoted as $S^u = (S_1^u, S_2^u, \ldots, S_{|S^u|}^u)$, where $S_t^u \in I$, and $1 \leq t \leq |S^u|$ is the time step. As for the sequential recommendation task, given the input as $(S_1^u, S_2^u, \ldots, S_{|S^u|-1}^u)$, our model will predict the next item rely on the previous t times at t time step. That is to say, the desired outputs at each time step are $(S_2^u, S_3^u, \ldots, S_{|S^u|}^u)$.

Table 1
Notations and their descriptions used in this paper.

Notations	Descriptions
Y	User-item interaction matrix
U	User set
I	Item set
G^{kg}	Knowledge graph
G^{Y}	User-item interaction bipartite graph
G	Integrated graph
S^u	User u's sequential behaviors
P	A high-order path
$P(u, i_1)$	High-order paths set between the (u, i_1) pair
$\mathbf{Interest_{seq}}$	Sequential-aware interest
$\mathbf{Interest_{KG}}$	Knowledge-aware interest
$\mathbf{A} \in \mathbb{R}^{\lvert I \rvert \times d}$	Item embedding matrix
$\mathbf{D} \in \mathbb{R}^{L \times d}$	Input item embedding matrix
$\mathbf{POS} \in \mathbb{R}^{L \times d}$	Positional embedding matrix

Formally, we use a triplet (h, r, t) to represent a fact in the KG, i.e., there is a relation r between the head entity h and tail entity t. We also denote the KG as $G^{kg} = \left\{(h, r, t) | h, t \in \varepsilon', r \in R'\right\}$, where ε' and R' are the entity set and relation set respectively. Likewise, the user interaction matrix can be regarded as a bipartite graph, which can be further represented as $G^Y = \{(u, interact, i) | u \in U, i \in \varepsilon'\right\}$. Since there is a consistent one-to-one match between each item and each entity in most cases, it is necessary for us to leverage this information to bridge the G^{kg} and G^Y. To integrate these two graphs, the operation we used is "entity alignment". Specifically, an item-entity map, $H = \left\{(i, e) | i \in I, \ e \in \varepsilon'\right\}$, is built for collecting the alignment information between each entity and each item. Finally, we can obtain a new integrated KG, which can be represented as $G = \{(h, r, t) \, | h, t \in \varepsilon, \ r \in R\}$, where $\varepsilon = \varepsilon' \cup U$ and $R = R' \cup \{interact\}$.

Furthermore, with the help of the KG, it is possible for each user-item pair to generate several high-order paths. Given a user u and an item i_1, a path connected between them can be expressed as $P = \left\{u \overset{r_1}{\rightarrow} e_1 \overset{r_2}{\rightarrow} \ldots \overset{r_i}{\rightarrow} i_1\right\}$. If there are l paths between u and i_1, all of them can be expressed as $P(u, i_1) = \{P_1, P_2, \ldots, P_l\}$.

The notations and their descriptions used in this paper are shown in Table 1.

3.1. Sequential-aware interest module

As aforementioned, we argue that user's preferences are actually complex and multidimensional. Therefore, these fine-grained preferences are modeled as intrinsic interests and external potential interests. In this module, the user's intrinsic interests are purely characterized via his/her sequential behaviors. Specifically, the given training sequence $(S_1^u, S_2^u, \ldots, S_{|S^u|-1}^u)$ should be converted to a fixed-length data $s = (i_1, i_2, \ldots, i_L)$, where L is the maximum length of a sequence. On the one hand, if the original sequence length is greater than L, we only select the most recent L items as the training data. On the other hand, the "pad" items are adopted to fill the sequence until its length is less than L. According to these training sequences, the user's sequential-aware interest can be denoted as follows:

$$\mathbf{Interest_{seq}} = f(s) \tag{2}$$

where $f(\cdot)$ is the learning function used for obtaining user's sequential-aware interest $\mathbf{Interest_{seq}}$ from the input data s.

3.2. Knowledge-aware interest module

On the other hand, user's preferences are actually dynamic and changeable, it is difficult to fully explore user's preferences by solely relying on their sequential behavior. To more completely

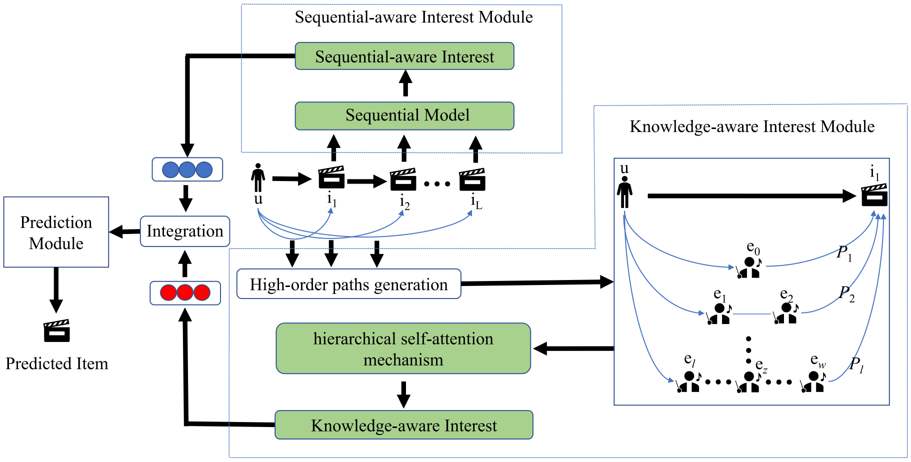
Sequential-aware Interest Module
Sequential-aware Interest
Sequential Model
u
i1
i2
iL
Prediction Module
Integration
High-order paths generation
hierarchical self-attention mechanism
Knowledge-aware Interest
Predicted Item
Knowledge-aware Interest Module
u
i1
e0
P1
e1
e2
P2
el
ez
ew
Pl

Fig. 1. The overall framework of GFE.

explore user's preferences, applying the KG is a promising strategy. This idea has several noteworthy benefits: (1) it is useful for recommendation to find out user's external potential preferences by introducing external knowledge for items. (2) After applying the KG, it is possible for each user-item pair to generate several high-order paths, which can be utilized to capture user's dynamic preference evolution from the high-order semantic information among entities. (3) the KG can provide high-order paths for the reasoning of the explainable recommendation model. In the light of these ideas, we develop a knowledge-aware interest module to investigate the utility of the KG for sequential recommendation.

In the knowledge-aware interest module, given pairs (u, i_1), $(u, i_2), \ldots, (u, i_L)$, the high-order paths generation is design for each of them to generate possible high-order paths with the help of the KG. Specifically, two entities can be connected by different relations, which can further become high-order paths with different semantic for user interactions. As show in Fig. 1, the u and i_1 can be connected by different paths, such as $P_1 = \left\{u \xrightarrow{interact} e_0 \xrightarrow{r_0} i_1\right\}, P_2 = \left\{u \xrightarrow{interact} e_1 \xrightarrow{r_2} e_2 \xrightarrow{r_i} i_1\right\}$ and so on. After obtaining these generated paths $P(u, i_1)$, $P(u, i_2), \ldots, P(u, i_L)$, the user's knowledge-aware interest can be denoted as follows:

$$\mathbf{Interest_{KG}} = g\left(P\left(u, i_1\right), P\left(u, i_2\right), P\left(u, i_L\right)\right) \tag{3}$$

where $g(\cdot)$ is another function for obtaining user's knowledge-aware interest $\mathbf{Interest_{KG}}$.

3.3. Prediction module

As aforementioned, both of the sequential-aware and knowledge-aware interests can be combined to describe user's fine-grained preferences and dynamic preferences evolution. That is to say, the sequential-aware preference representation together with the knowledge-aware preference representation are combined as the final representation of user's preference. To take the advantage of these two interests, the integration of them can be denoted as follows:

$$\mathbf{Interest} = \varphi\left(W\left[\mathbf{Interest_{seq}}, \mathbf{Interest_{KG}}\right] + b\right) \tag{4}$$

where φ is the non-linear activation function, W and b are the weights and bias parameters respectively, [,] is the concatenation operation, **Interest** is the integrated interest.

Furthermore, the predicted output of item i would be calculated via a latent factor model as follows:

$$r_{i,t} = \sigma\left(\mathbf{Interest}_t \mathbf{E}_i\right) \tag{5}$$

where $\mathbf{Interest}_t$ is the representation of user's interests given the first t time, i.e., $(i_1, i_2, \ldots, i_t)$ and $\mathbf{E}_i$ is the embedding of the item i, σ is the sigmoid function to convert the score range to (0,1).

As aforementioned, we use "pad" item to fill the sequence when its length is less than L, so its desired output is also "pad". We use a negative sample strategy to optimize our model. Specifically, we sample a negative item i^{neg} for each desired positive output i^{pos}. The binary cross entropy loss is applied as the objective function of our model, which can be denoted as follows:

$$\zeta = -\sum_{S^u \in S} \sum_{t \in [1,2,\ldots,L]} \left[\log\left(r_{i^{neg}}, t\right) + \log\left(r_{i^{pos}}, t\right)\right] + \lambda \|\theta\|_F^2 \tag{6}$$

where $\|\theta\|_F^2$ is the L2 regularization with parameter λ to prevent overfitting..

4. Recommendation models under framework

In order to show the strong availability and generalization of GFE, we propose two extended models based on the GFE and two pure sequential models (i.e., SASRec [13] and TiSASRec [14]), termed as GFE-SASRec and GFE-TiSASRec. Note that SASRec and TiSASRec are two state-of-the-art sequential recommendation models that are based on the self-attention mechanism.

4.1. GFE-SASRec model

As shown in Fig. 2, we propose a model to integrate the SASRec with GFE, i.e., GFE-SASRec. In the GFE-SASRec, we first extend the sequential-aware interest module. Specifically, an embedding matrix $\mathbf{A} \in \mathbb{R}^{|I| \times d}$ is built for all items, where d is the hidden dimensionality. Likewise, the input embedding can be retrieved from $\mathbf{A}$ via the look-up operation as $\mathbf{D} \in \mathbb{R}^{L \times d}$. Note that we employ the constant zero vector for the padding item's embedding. In addition, as shown in Fig. 2, self-attention is used for adaptively capture the important sequential patterns. However, it does not contain convolutional or recurrent operations, resulting in a lack of position awareness. To extend its position-aware ability, a position embedding $\mathbf{POS} \in \mathbb{R}^{L \times d}$ is employed to the $\mathbf{D}$ via the concatenation operation, which can be denoted as follows:

$$\mathbf{J} = \begin{bmatrix} \mathbf{D}_{i_1} + \mathbf{POS}_1 \\ \mathbf{D}_{i_2} + \mathbf{POS}_2 \\ \cdots \\ \mathbf{D}_{i_L} + \mathbf{POS}_L \end{bmatrix} \tag{7}$$

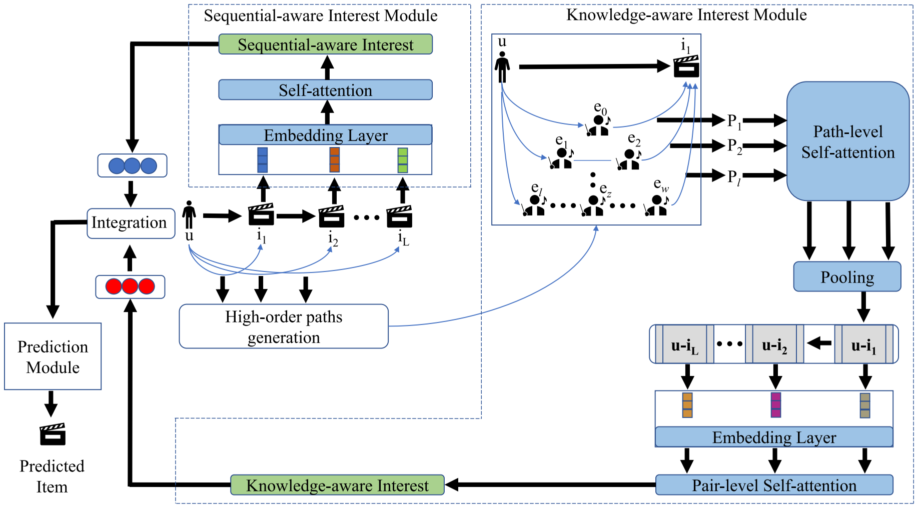

Fig. 2. The overall framework of GFE-SASRec.

where $\mathbf{D}_{i_L}$ and $\mathbf{POS}_L$ are the Lth item embedding and position embedding respectively.

Furthermore, the extended embedding will be fed into the self-attention layer to adaptively calculate the importance of different items for the user's preferences. The self-attention operation can be expressed as follows:

$$\mathbf{Q} = \mathbf{E}W^Q \tag{8}$$

$$\mathbf{K} = \mathbf{E}W^K \tag{9}$$

$$\mathbf{V} = \mathbf{E}W^V \tag{10}$$

$$\mathbf{S}_t = \text{self-attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \tag{11}$$

$$= \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V} \tag{12}$$

where $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are the queries, keys, values, $W^Q \in \mathbb{R}^{d\times d}$, $W^K \in \mathbb{R}^{d\times d}$ and $W^V \in \mathbb{R}^{d\times d}$ are three matrices for the $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ to project respectively, $\sqrt{d}$ is the scale factor, $1 \leq t \leq L$.

Considering that the self-attention layer is still linear, we endow non-linearity to it to capture complex user-item interaction. To achieve this goal, a point-wise feed-forward network is employed as follows:

$$\mathbf{F}_t = FFN\left(\mathbf{S}_t\right) = ReLU\left(\mathbf{S}_t W^t + b^t\right) W^{t_1} + b^{t_1} \tag{13}$$

where $ReLU$ is the non-linear activation function, $W^t \in \mathbb{R}^{d\times d}$ and $W^{t_1} \in \mathbb{R}^{d\times d}$ are the weights parameters, $b^t \in \mathbb{R}^d$ and $b^{t_1} \in \mathbb{R}^d$ are the bias parameters. Besides, stacking the self-attention layer is a promising strategy to improve the model's performance. However, it is worth noting that as the number of stacked layers increases, it may also bring some potential problems, such as overfitting, harder to train and so on. Follow the [12–14], we adopt the layer normalization, dropout and residual connections techniques to alleviate these limitations. Specifically, as for each layer, we utilize the layer normalization method for the layer input to normalize its across features. Next, the dropout strategy is adopted for the layer output, which can prevent overfitting. Finally, residual connections would be leveraged to propagate features from bottom layers to higher layers, so that more useful features can be hierarchically learnt via these residual stacks. After that, the above steps can be expressed as follows:

$$\mathbf{F}^{\mathbf{c}} = \text{Self-attention}\ ^c(\mathbf{J}) \tag{14}$$

where Self-attention $^c(\cdot)$ is the self-attention operation with c layers, $\mathbf{F}^{\mathbf{c}}$ is the output after stacking c layers self-attention. In addition, $\mathbf{F}^{\mathbf{c}}$ is also regarded as the representation of user's sequential-aware interest, so Eq. (1) can be rewritten as follows:

$$\mathbf{Interest_{seq}} = f(s) = \mathbf{F}^{\mathbf{c}} \tag{15}$$

Moreover, GFE-SASRec extends the knowledge-aware interest module by proposing a hierarchical self-attention mechanism, which can adaptively characterize the weights of different items and further aggregate their sequential semantic signals better. The hierarchical self-attention mechanism contains two level strategies, called path-level self-attention and pair-level self-attention. Thanks for the KG, it is helpful for recommendation to generate high-order paths for each user-item pair. Given a (u, i_1) pair and its generated paths $P(u, i_1) = \{P_1, P_2, \ldots, P_l\}$, it is intuitive for us to regard the paths as a sequence, so a path-level self-attention is employed to capture their features. Specifically, taking the $P = \left\{u \xrightarrow{r_1} e_1 \xrightarrow{r_2} e_2 \ldots \xrightarrow{r_i} i_1\right\}$ as an example, we concatenate each entity and its connected relation as a new input, which can be represented as follows:

$$\mathbf{Z}_1 = \left[\mathbf{u}_1, \mathbf{r}_1\right], \mathbf{Z}_2 = \left[\mathbf{e}_1, \mathbf{r}_2\right], \ldots, \mathbf{Z}_{i_1} = \left[\mathbf{i}_1, \textbf{"null"}\right] \tag{16}$$

where $\mathbf{u}_1, \mathbf{r}_1,\ \mathbf{e}_1,\ \mathbf{r}_2, \ldots,\ \mathbf{i}_1$ are the embeddings of $u_1, r_1,\ e_1, r_2, \ldots,\ i_1$ respectively, [,] is the concatenation operation. Note that "null" is the padding relation for i_1. Likewise, these inputs will further feed into the path-level self-attention as follows:

$$\mathbf{P} = \ \text{Self-attention}\ ^c\left(\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_{i_1}\right) \tag{17}$$

where $\mathbf{P}$ is the representation of the path P. Similarly, after repeating the above path-level self-attention operation, we can obtain total representations of these l paths between the (u, i_1) pair, i.e., $\mathbf{P}_1,\ \mathbf{P}_2, \ldots,\ \mathbf{P}_l$. Furthermore, different paths have different high-order semantic information for user's dynamic preference evolution, it should aggregate them with different weights. Here we adopt a pooling strategy based on attention mechanism to obtain the aggregation representation of paths between (u, i_1) pair:

$$\mathbf{u}{-}\mathbf{i}_1 = \sum_{h=1}^{l} a_h \cdot \mathbf{P}_h \tag{18}$$

Likewise, we can collect other aggregation representations for user-item pairs, such as $\mathbf{u}{-}\mathbf{i}_2,\ \mathbf{u}{-}\mathbf{i}_3, \ldots,\ \mathbf{u}{-}\mathbf{i}_{\mathrm{L}}$. As for these representations, they are also can be regarded as a sequence $(\mathbf{u}{-}\mathbf{i}_1,\ \mathbf{u}{-}\mathbf{i}_2, \ldots,\ \mathbf{u}{-}\mathbf{i}_{\mathrm{L}})$. Such sequence contains semantic information of user's preference evolution. Therefore, we design a pair-level self-attention to characterize them and obtain the user's knowledge-aware interest, so Eq. (2) can be rewritten as follows:

$$\textbf{Interest}_{\mathrm{KG}} = g\left(P\left(u, i_1\right), P\left(u, i_2\right), P\left(u, i_{\mathrm{L}}\right)\right) \tag{19}$$

$$= \text{Self-attention}^{\,c}\left(\mathbf{u}{-}\mathbf{i}_1, \mathbf{u}{-}\mathbf{i}_2, \ldots, \mathbf{u}{-}\mathbf{i}_{\mathrm{L}}\right) \tag{20}$$

After finishing that, we obtain the user's sequential-aware interest and knowledge-aware interest from two modules, which can be further fed into the prediction module.

4.2. GFE-TiSASRec model

As shown in Fig. 3, we propose another model to integrate the TiSASRec with GFE, i.e., GFE-TiSASRec. Since the knowledge-aware interest module in the GFE-TiSASRec is the same as the knowledge-aware interest module used in the GFE-SASRec, so we only expound the sequential-aware interest module designed in the GFE-TiSASRec. Given the training sequence s and its time sequence $T_u = (t_1, t_2, \ldots, t_L)$, the time intervals between two items can be computed as $t_n - t_m$ $(n > m)$ and the time interval set of u is built as $\mathbf{N_u}$. The minimum element in $\mathbf{N_u}$ would be represented as $N^u_{min} = min(\mathbf{N_u})$, thus the scaled interval can be calculated as $N^u_{mn} = \left\lfloor \frac{t_n - t_m}{N^u_{min}} \right\rfloor$ $(n > m)$. After finishing that, a time interval matrix can be built as follows:

$$\mathbf{M_u} = \begin{bmatrix} N^u_{11} & N^u_{12} & \ldots & N^u_{1L} \\ N^u_{21} & N^u_{22} & \ldots & N^u_{2L} \\ \ldots & \ldots & \ldots & \ldots \\ N^u_{L1} & N^u_{L2} & \ldots & N^u_{LL} \end{bmatrix} \tag{21}$$

where $\mathbf{M_u}$ is a symmetric matrix whose main diagonal elements are all 0. In addition, TiSASRec adopts a clip operation for $\mathbf{M_u}$ to decrease its elements' value. Specifically, for each element in $\mathbf{M_u}$, N^u_{mn} will be changed as a predetermined certain threshold k if its value is greater than k.

In the GFE-TiSASRec, the embedding matrix $\mathbf{A} \in \mathbb{R}^{|I| \times d}$ and retrieval input embedding $\mathbf{D} \in \mathbb{R}^{L \times d}$ are also created. Since the position-aware ability is also vital for the TiSASRec, positional embedding is adopted to concatenate with the input retrieval embedding. To achieve this goal, follow the [14] and [37], we build two positional embeddings $\mathbf{POS^k} \in \mathbb{R}^{L \times d}$ and $\mathbf{POS^v} \in \mathbb{R}^{L \times d}$ for the concatenation of keys and values in the self-attention respectively, which can be denoted as follows:

$$\mathbf{POS^k} = \begin{bmatrix} \mathrm{POS}^k_1 \\ \mathrm{POS}^k_2 \\ \ldots \\ \mathrm{POS}^k_{\mathrm{L}} \end{bmatrix} \tag{22}$$

$$\mathbf{POS^v} = \begin{bmatrix} \mathrm{POS}^v_1 \\ \mathrm{POS}^v_2 \\ \ldots \\ \mathrm{POS}^v_{\mathrm{L}} \end{bmatrix} \tag{23}$$

Moreover, we create two relative time interval embeddings $\mathbf{TI^k}$ and $\mathbf{TI^v}$ for the concatenation of keys and values respectively, which can be denoted as follows:

$$\mathbf{TI^k} = \begin{bmatrix} N^k_{11} & N^k_{12} & \ldots & N^k_{1L} \\ N^k_{21} & N^k_{22} & \ldots & N^k_{2L} \\ \ldots & \ldots & \ldots & \ldots \\ N^k_{L1} & N^k_{L2} & \ldots & N^k_{LL} \end{bmatrix} \tag{24}$$

$$\mathbf{TI^v} = \begin{bmatrix} N^v_{11} & N^v_{12} & \ldots & N^v_{1L} \\ N^v_{21} & N^v_{22} & \ldots & N^v_{2L} \\ \ldots & \ldots & \ldots & \ldots \\ N^v_{L1} & N^v_{L2} & \ldots & N^v_{LL} \end{bmatrix} \tag{25}$$

After finishing that, we use the time interval-aware self-attention to combine the above embeddings follow the TiSASRec. Given the input $\mathbf{D}$, the calculation for obtaining the output $\mathbf{S}_t$ can be denoted as follows:

$$a_{mn} = \frac{\exp \frac{\mathbf{D}_{i_m} W^Q \left(\mathbf{D}_{i_n} W^K + N^k_{mn} + \mathrm{Pos}^k_n\right)^T}{\sqrt{d}}}{\sum_{h=1}^{L} \exp \frac{\mathbf{D}_{i_m} W^Q \left(\mathbf{D}_{i_h} W^K + N^k_{mh} + \mathrm{POS}^k_h\right)^T}{\sqrt{d}}} \tag{26}$$

$$\mathbf{S}_t = \sum_{n=1}^{L} a_{mn}\left(\mathbf{D}_{i_n} W^V + N^v_{mn} + \mathrm{POS}^v_n\right) \tag{27}$$

where W^Q, W^K and W^V are the projection matrices for the queries, keys, and values respectively.

Likewise, we also adopt the non-linear activation function, layer normalization, dropout and residual connections techniques to stack the self-attention follow the Equations from (13) to (15).

5. Experiments

In this section, we evaluate the performance of GFE on three real-world public datasets: Movielens-100k and Movielens-1M and Amazon Book. We first introduce the compared baselines and experiment settings. Then, we compare the GFE-based models with state-of-the-art baselines and discuss how different parameters impact the model performances. Finally, we give a case study to demonstrate how GFE can offer reasonable explanations with respect to the user's preferences on items.

5.1. Datasets & baselines

The datasets we adopted are described as follows:

- **Movielens-1M**[1]: A movie dataset that contains about 1 million ratings (from 1 to 5. "5" stands for "the best rating" while "1" stands for "the lowest rating").
- **Movielens-100k**[2]: This is a smaller movie dataset than Movielens-1M that comprises around 100,000 ratings given by 1000 users on 1700 movies (from 1 to 5. "5" stands for "the best rating" while "1" stands for "the lowest rating").
- **Amazon Book** [38]: It is a larger dataset collected by the Amazon website, which contains the user's ratings on book products.

Considering that sequential recommendation puts more emphasis on exploring user's interests from implicit feedback data elaborated in Section 3, all the above datasets should be converted to implicit feedback data. Specifically, all the observed interactions are regarded as positive feedback while the unobserved interactions are negative feedback.

IMDB[3] is a useful tool to construct movie knowledge graph, which contains abundant auxiliary information of the movies we used. In this step, we utilize the titles and release dates of movies in the Movielens-1M/Movielens-100k to link to the IMDB. As mentioned in Section 3, we apply the "entity alignment" operation to integrate Movielens-1M/Movielens-100k with IMDB. Similarly, as for the Amzon Book dataset, we adopt the KB4Rec dataset [39] to construct an aligned link between KG information and user interaction data. Finally, the detailed basic statistics of the three datasets are shown in Table 2.

To evaluate the performance of GFE-based models, i.e., GFE-SASRec and GFE-TiSASRec, we choose 9 state-of-the-art models as the baselines which are described in detail as follows:

[1] https://grouplens.org/datasets/movielens/1m/.

[2] https://grouplens.org/datasets/movielens/100k/.

[3] http://www.imdb.com/.

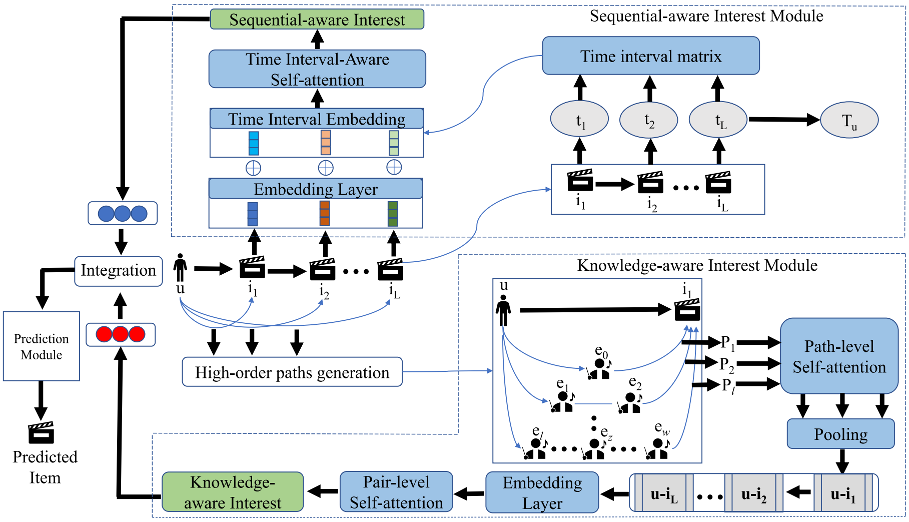

Fig. 3. The overall framework of GFE-TiSASRec.

Table 2
Basic statistics of the datasets.

	Movielens-100k	Movielens-1M	Amazon Book
User	943	6040	75645
Item	1347	3167	26144
Average length	103.27	155.47	10.09
Relation types	3	3	38

- **BPR** [40]: It is a factorization machine model based on feature interactions which is used widely in CTR scenario. In our experiments, the user ID and item ID are concatenated as input.
- **NCF** [41]: It is a collaborative filtering based model that applies neural network instead of the inner product to improve the performance of recommendation.
- **Tensor** [42]: It is a collaborative filtering method based on tensor factorization, which allows a flexible and generic integration of contextual information and further can be applied to offer context-aware recommendation.
- **GRU4Rec** [7]: It is a classical sequential recommendation model that adopts GRU to model user's sequence behaviors.
- **GRU4Rec+KG:** It is an extension of GRU4Rec. We implement it by pre-training the item embeddings with the help of the KG. And the pre-trained KG embedding of item will be treated as feature vector for the GRU4Rec's input.
- **NARM** [8]: It is RNN-based sequential recommendation method that combines the power of the GRU and attention mechanism, which can capture user's long-term preferences.
- **KSR** [35]: It is a knowledge-enhanced sequential recommendation, which proposed GRU-based networks with Key-Value Memory Network (KV-MN). The knowledge information is utilized to enhance the semantic representation of KV-MN.
- **SASRec** [13]: It is self-attention based model that significantly outperforms state-of-the-art Markov Chain/CNN/RNN-based sequential recommendation methods without any convolutional or recurrent operation.
- **TiSASRec** [14]: It is another state-of-the-art sequential recommendation model based self-attention, which extend the SASRec by considering time interval information.

Table 3
Overall Performance Comparison on Movielens-100k.

	Movielens-100k			
	HR@5	NDCG@5	HR@10	DCG@10
BPR	0.4104	0.2755	0.5705	0.3290
NCF	0.4199	0.2952	0.5726	0.3440
Tensor	0.3690	0.2487	0.5419	0.3041
GRU4Rec	0.4613	0.3143	0.6329	0.3698
GRU4Rec+KG	0.4801	0.3254	0.6462	0.3776
NARM	0.4889	0.3279	0.6511	0.3799
KSR	0.5279	0.3571	0.6849	0.4166
SASRec	0.5500	0.3894	0.7292	0.4422
TiSASRec	0.5610	0.4011	0.7333	0.4570
GFE-SASRec	**0.5726**	**0.4137**	**0.7432**	**0.4681**
GFE-TiSASRec	**0.5793**	**0.4206**	**0.7511**	**0.4766**

5.2. Experiment setting

We firstly group the interactions by users and sort the sequence order of items by comparing their timestamps. After that, we remove cold-start users/items with fewer than 5 behaviors. As for the alignment of Amazon Book dataset, we also remove entities with fewer than 3 KG triples to improve the KG quality. To evaluate the sequential recommendation, we apply the leave-one-out method to split the dataset, which is widely used in [4,41,43]. Specifically, we use the most recent item as test data, the item just before the last item as validation set and the remaining items as training data. To evaluate the performance of all the models, we adopt four common metrics, including Hit Ratio @5 (HR@5 for short), Hit Ratio @10 (HR@10 for short), Normalized Discounted Cumulative Gain @5 (NDCG@5 for short) and Normalized Discounted Cumulative Gain @10 (NDCG@10 for short). Hit Ratio metrics are used to calculate the rate of items that the ground-truth items among the top 5/10 items, while the Normalized Discounted Cumulative Gain metrics are utilized to endow high weights to higher positions. To improve the computation efficiency, we use the negative sampling strategy. We randomly sample 100 negative items for each user, and pair the ground-truth item in the test data with these sample items for evaluation. That is to say, all the evaluation metrics will be computed according to the rankings of these 101 items. During the model training, we utilized the Adam algorithm [44] to optimize parameters. The more settings would be discussed in Section 5.4.

Table 4
Overall Performance Comparison on Movielens-1M.

	Movielens-1M			
	HR@5	NDCG@5	HR@10	DCG@10
BPR	0.4835	0.3301	0.6535	0.3882
NCF	0.4847	0.3409	0.6599	0.3917
Tensor	0.4503	0.3077	0.6320	0.3666
GRU4Rec	0.6098	0.4361	0.7550	0.4881
GRU4Rec+KG	0.6175	0.4474	0.7624	0.4957
NARM	0.6103	0.4407	0.7556	0.4904
KSR	0.6879	0.5194	0.7975	0.5661
SASRec	0.7243	0.5655	0.8286	0.5996
TiSASRec	0.7317	0.5750	0.8288	0.6068
GFE-SASRec	**0.7411**	**0.5871**	**0.8367**	**0.6173**
GFE-TiSASRec	**0.7473**	**0.5950**	**0.8431**	**0.6267**

Table 5
Overall Performance Comparison on Amazon Book.

	Amazon Book			
	HR@5	NDCG@5	HR@10	DCG@10
BPR	0.5777	0.4305	0.7113	0.4738
NCF	0.5866	0.4511	0.6993	0.4877
Tensor	0.5704	0.4405	0.6804	0.4762
GRU4Rec	0.6047	0.4597	0.7317	0.5008
GRU4Rec+KG	0.6177	0.4709	0.7429	0.5169
NARM	0.5997	0.4558	0.7255	0.4966
KSR	0.6451	0.5196	0.7677	0.5591
SASRec	0.6774	0.5418	0.7812	0.5757
TiSASRec	0.6836	0.5481	0.7898	0.5808
GFE-SASRec	**0.7001**	**0.5644**	**0.8051**	**0.6021**
GFE-TiSASRec	**0.7075**	**0.5701**	**0.8114**	**0.6102**

5.3. Performance comparison

In this section, we compare GFE-SASRec and GFE-TiSASRec with the 9 state-of-the-art models in sequential recommendation task. The results of all the models are shown in Table 3, Tables 4 and 5 respectively. And we have the following observations:

- GFE-based models, i.e., GFE-SASRec and GFE-TiSASRec, perform best on three datasets. The results reveal that they can have a better ability to capture user's fine-grained preferences and their dynamic preferences evolution. Specifically, the sequential-aware interest module and knowledge-aware interest can play a positive role on exploring user's intrinsic interests and external potential interests, thus providing better recommendation results.
- By comparing the results of GFE-SASRec and SASRec, and GFE-TiSASRec and TiSASRec, GFE-based models outperform them. It shows that GFE is able to utilize the high-order semantic relevance among entities from the KG. That is to say, GFE-based models can perform better by benefiting from the utility of the KG instead of solely utilizing the sequence interaction. Besides, the performances of the SASRec and TiSASRec are improved through integrating the GFE, which demonstrates that GFE is helpful for sequential recommendation model to capture the transition patterns of user's sequence.
- The results of SASRec, TiSASRec, GFE-SASRec and GFE-TiSASRec outperform other baselines significantly. A vital factor is that all of them are based on the self-attention mechanism, which demonstrates that this mechanism is capable to capture sequential patterns well.
- The results also show that other sequential recommendation models, such as GRU4Rec and NARM, have better performances than the general recommendation models, such as BPR and NCF. The reason is that the general recommendation models focus only on the user-item interaction information, while the sequential recommendation models can be enhanced by the sequential patterns.
- After integrating the KG information to enhance the representation of item, the GRU4Rec's performance has been improved. Besides, KSR also performs satisfactory results among all datasets. It utilized the KG-enhanced KV-MN to capture attribute-level user's preference, which can be combined with the sequential preference representation for the improvement of recommendation performance. In a word, KG information is a useful information for the sequential recommendation.

5.4. Ablation analysis

In GFE framework, there are some important components, e.g., sequential-aware module, knowledge-aware module and hierarchical self-attention mechanism. In this experiment, we conduct an ablation analysis on these important components in order to better understand their effects on the model performance. The results are reported on Tables 6 and 7. Firstly, Since the sequential-aware module is necessary for the sequential recommendation, we remove the knowledge-aware module to obtain its variants. Through comparing the results before and after removing the knowledge-aware module, it is obvious that the GFE-SASRec and GFE-TiSASRec have better performances. It demonstrates that this module can improve sequential recommendation performance via capturing user's external potential interests. That is to say, the combination of sequential-aware preference representation together with the knowledge-aware preference representation can beneficial for the final representation of user's preference. Besides, in the knowledge-aware interest module, we propose a hierarchical self-attention mechanism, i.e., path-level self-attention and pair-level self-attention, to aggregate the high-order semantic information from high-order paths. The results show that both the GFE-SASRec and GFE-TiSASRec are capable to benefit from the hierarchical self-attention mechanism. Because both of them perform worse without the path-level self-attention or pair-level self-attention. In addition, both the GFE-SASRec and GFE-TiSASRec perform worse without path-level self-attention than without pair-level self-attention. It indicates that the path-level plays a more important role on aggregating high-order semantic information. We argue that this may be because the path-level self-attention can learn more abundant semantic information from different entities and paths than from the pair-level information.

5.5. Parameters study

In this section, we develop several sensitivity experiments to explore the effects of different parameters on the model performances, such as the hidden dimensionality, learning rate, sequence length and self-attention layer. The analyses are conducted on the two datasets via a variable-controlling approach, i.e., only the analyzed variable is changed while other parameters in the model are the same.

5.5.1. Parameter analysis on the hidden dimensionality

We adjust the number of hidden dimensionalities from {10,20,30,40,50,60,70,80} to investigate the impact of d on the performance of the models. From Fig. 4, we find that GFE-SASRec and GFE-TiSASRec perform consistently better than other baselines in all the hidden dimensionalities. To some extent, the larger hidden dimensionality is helpful to improve the performance of models. However, when the hidden dimensionality has achieved a certain value, the performance of the model begins to decrease. Another observation is that all the model performances attend to converge as the dimensionality increases.

Table 6
Ablation analysis on Movielens-100k.

	Movielens-100k			
	HR@5	NDCG@5	HR@10	DCG@10
GFE-SASRec	0.5726	0.4137	0.7432	0.4681
GFE-SASRec w/o Knowledge-aware module	0.5500	0.3844	0.7292	0.4422
GFE-SASRec w/o Path-level Self-attention	0.5627	0.4031	0.7355	0.4603
GFE-SASRec w/o Pair-level Self-attention	0.5667	0.4079	0.7395	0.4641
GFE-TiSASRec	0.5813	0.4279	0.7521	0.4766
GFE-TiSASRec w/o Knowledge-aware module	0.5610	0.4011	0.7333	0.4570
GFE-TiSASRec w/o Path-level Self-attention	0.5748	0.4152	0.7449	0.4692
GFE-TiSASRec w/o Pair-level Self-attention	0.5779	0.4203	0.7481	0.4711

Table 7
Ablation analysis on Movielens-1M.

	Movielens-1M			
	HR@5	NDCG@5	HR@10	DCG@10
GFE-SASRec	0.7411	0.5872	0.8367	0.6172
GFE-SASRec w/o Knowledge-aware module	0.7243	0.5655	0.8286	0.5996
GFE-SASRec w/o Path-level Self-attention	0.7269	0.5687	0.8255	0.6013
GFE-SASRec w/o Pair-level Self-attention	0.7295	0.5724	0.8272	0.6039
GFE-TiSASRec	0.7473	0.5960	0.8431	0.6256
GFE-TiSASRec w/o Knowledge-aware module	0.7317	0.5750	0.8288	0.6068
GFE-TiSASRec w/o Path-level Self-attention	0.7397	0.5804	0.8321	0.6125
GFE-TiSASRec w/o Pair-level Self-attention	0.7434	0.5911	0.8389	0.6207

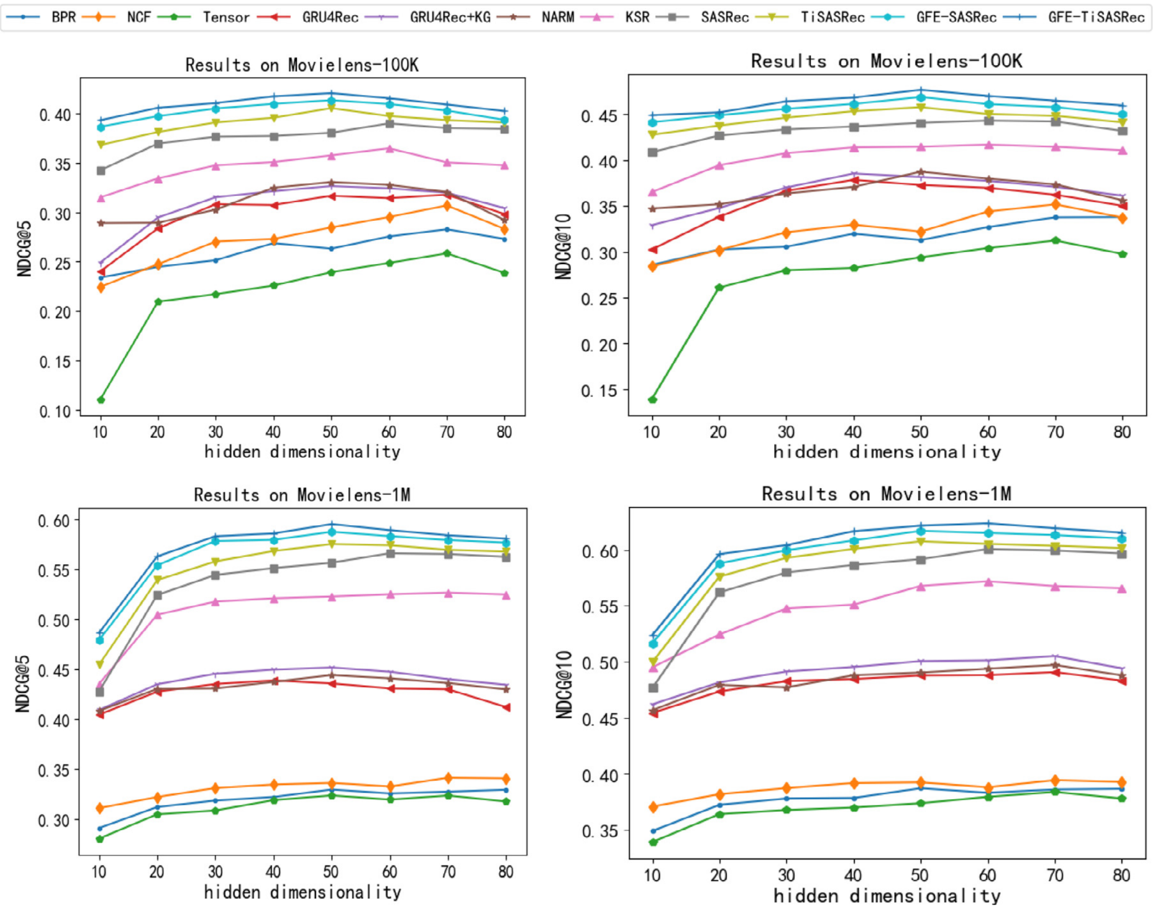

Fig. 4. The results of different hidden dimensionalities.

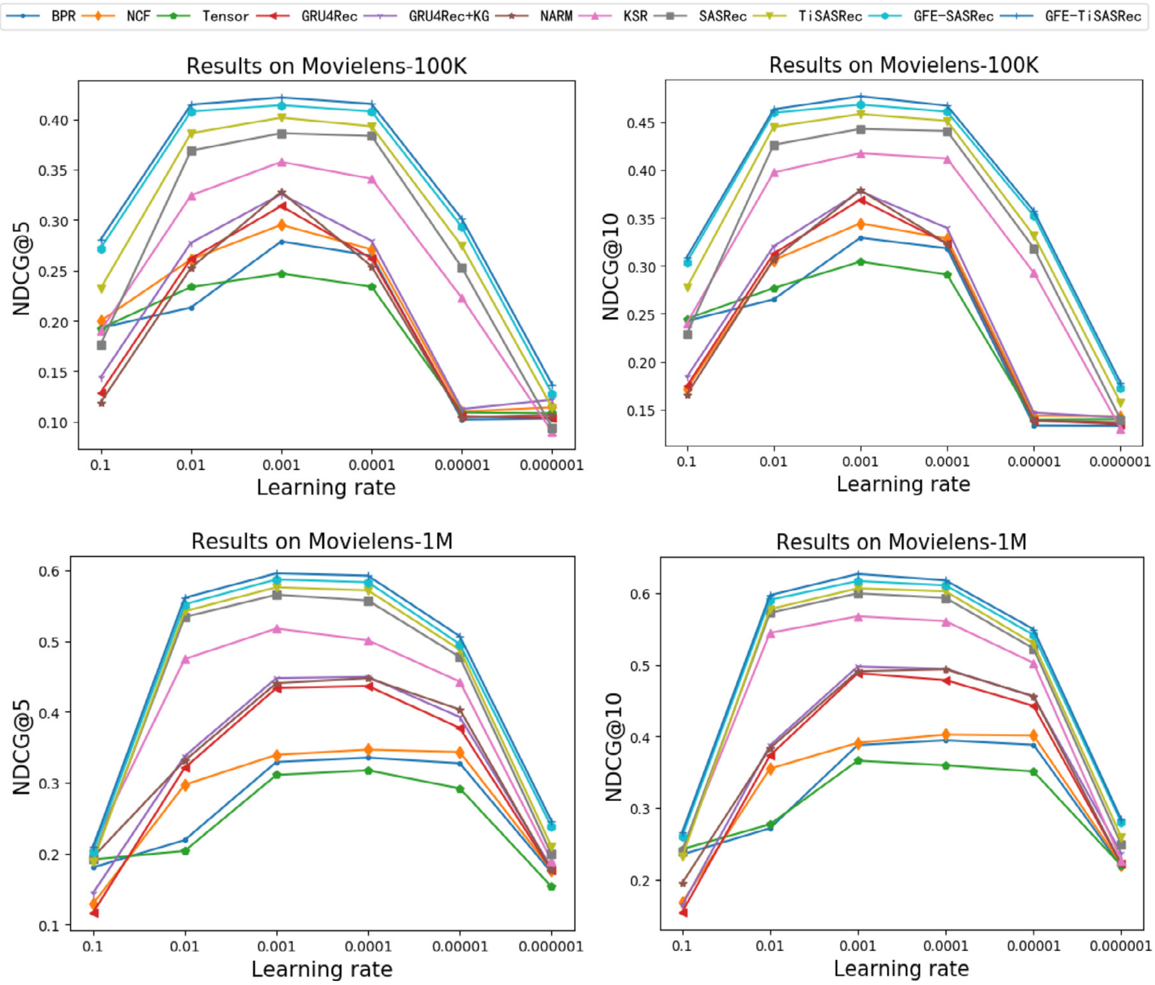

Fig. 5. The results of different learning rates.

5.5.2. *Parameter analysis on the learning rate*

We now investigate the effect of learning rate on the model performance. Fig. 5 shows the results of all models with the learning rate varying from 0.000001 to 0.1. The most obvious observation from these results is that a larger learning rate or smaller learning rate is not needful for better model performance. When the learning rate decreases to a certain extent, the model performances decrease sharply. Especially it is obvious that learning rate=0.001performs better than other learning rates.

5.5.3. *Parameter analysis on the sequence length*

We also study how the sequence length affects the model performance. The results with different sequence lengths on two datasets are shown in Fig. 6. Note that the larger sequence length is not adopted due to the memory limitation of our GPU. As for the sequential task, the sequence length is a vital impact that determines how many sequential signals that model can learn from. One observation is that all the model performances attend to rise as the sequence length increases. It reveals that they can benefit from a longer sequence. Another sight is that the growth rate of the models gradually decreases in the Movielens-1M dataset. Although the larger sequence length is able to provide more sequential patterns for recommendation, they may introduce unpredictable noise information.

5.5.4. *Parameter analysis on the self-attention layer*

As aforementioned, stacking self-attention layers is a promising strategy to improve the recommendation performance. As shown in Fig. 7, the results demonstrate that multiple self-attention is able to enhance the performances especially on the larger dataset (i.e., Movilens-1M). Within a certain range, the performance of the model increases with the increase of the number of self-attention layers. From Fig. 7, we find that the models perform better with 2 self-attention layers on the Movielens-100k dataset and with 4 self-attention layers on the Movilens-1M dataset. Besides, it is also obvious that stacked self-attention layers if help for capturing the complex sequential patterns. However, another observation is that too deeper self-attention layer will decrease the model performances, which shows that these models are overfitting.

5.6. *Analysis of explainability*

As aforementioned, the experiments have proven that the GFE can provide accurate recommendation results. Besides, GFE is able to offer reasonable explanations for users from two perspectives. Here, we select real examples from Movielens-1M to a visual impression of how GFE can provide explanations. We choose sequence behaviors of user $u62$ and $u101$ shown in Figs. 8 and 9 respectively (Note that the movie pictures are obtained from the IMDB). In these figures, the histograms are drawn according to the attention weights of different relation factors. The attention weights are adaptively calculated by the self-attention mechanism. The higher the attention mechanism score, the higher the height of the histogram, indicating that this factor contributes more to the user's preference. From the visualization, we have the following observations:

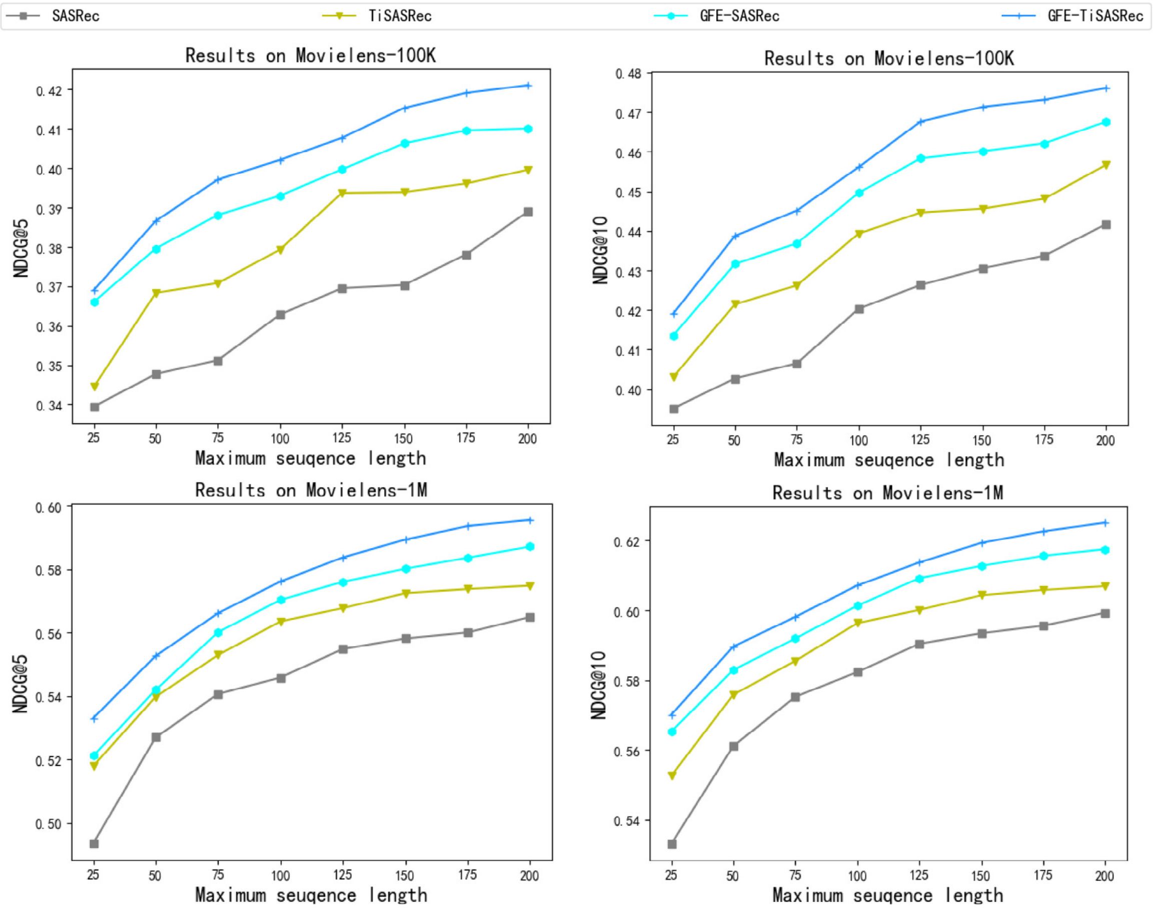

Fig. 6. The results of different maximum sequence lengths.

- For the microcosmic view (i.e., observing the knowledge relations from the vertical axis of the figure), it is obvious for us to understand the user's fine-grained preferences at each time step. For example, at $t1$, user $u62$ interacted with the *Prizzi's Honor* due to its genre factor, i.e., comedy (because the "genre" factor has the highest attention weight at this time step); the $u62$ liked the director of *Psycho* at $t2$, so that he/she decided to watch it; the reason why $u62$ chose the *Blade Runner* is that he/she has preferred the actors of this movie at $t4$. Similarly, as shown in Fig. 9, the reason why user $u101$ watched the *Get Shorty* at $t1$ step is that he/she liked this movie's actor. And he/she preferred the "action" movie during $t5$-$t6$. Another interesting observation is that user $u101$ interacted with *Beverly Hills Cop III* at $t3$, *Total Recall* at $t5$ and *Die Hard* at $t6$, although all of them are "action" movies, but $u101$ preferred the "director" of *Beverly Hills Cop III* at $t3$ rather than the "action" factor. According to these fine-grained observations, it is helpful for us to learn the reason for user's decision at a specific time.
- For the macroscopical view (i.e., observing the knowledge relations from the horizontal axis of the figure), we also can describe the user's dynamic preference evolution. $u62$ was influenced by the comedy movie at $t1$. Then he/she turned to watch the movie *Psycho* at $t2$ because he/she liked its director and watched the *Blade Runner* at t3 for its actors. At $t4$ step, he/she was driven by the sci-fi movie. During $t5$ and $t6$, he/she enjoyed the comedy movies again. From these evolutions, the user $u62$ seems to be a comedy fan. Similarly, we also can guess that the $u101$ seem to be an action movie fan.
- Base on the above discussion, the GFE can offer high-quality explanations from different views. That is to say, GFE can not only observe the fine-grained interests of users at a specific time step from a microcosmic perspective, but also see the dynamic evolution of user's interests over a period of time from a macroscopical perspective. For example, since genre factor, i.e., comedy, mainly drives the preferences of user $u62$, it is able to offer a recommendation *A Fish Called Wanda* to him/her at time $t7$. The explanation can be generated for $u62$ as: the *A Fish Called Wanda* is recommended since he/she has preferred the movies *Sleeper* and *Bananas* with the same genre, i.e., comedy. Similarly, the explanation can be generated for $u101$ as: the *I Love Trouble* is recommended since he/she has preferred the movies *Total Recall* and *Die Hard* with the same genre, i.e., action.

6. Conclusion and future work

In this paper, to break the limitations of traditional sequential recommendation, we propose a General Knowledge Enhanced Framework for Explainable Sequential Recommendation, called GFE, aiming at improving the accuracy and explainability of sequential recommendation. Firstly, GFE can capture user's fine-grained preferences by modeling them as intrinsic interests and external potential interests. Both of them can be obtained via the sequential-aware interest module and knowledge-aware interest module respectively. Moreover, it is equally important for sequential recommendation to explore user's dynamic preference evolution. It is possible to generate several high-order paths with the help of the KG for each user-item pair, which contains the high-order semantic relevance among items. Therefore,

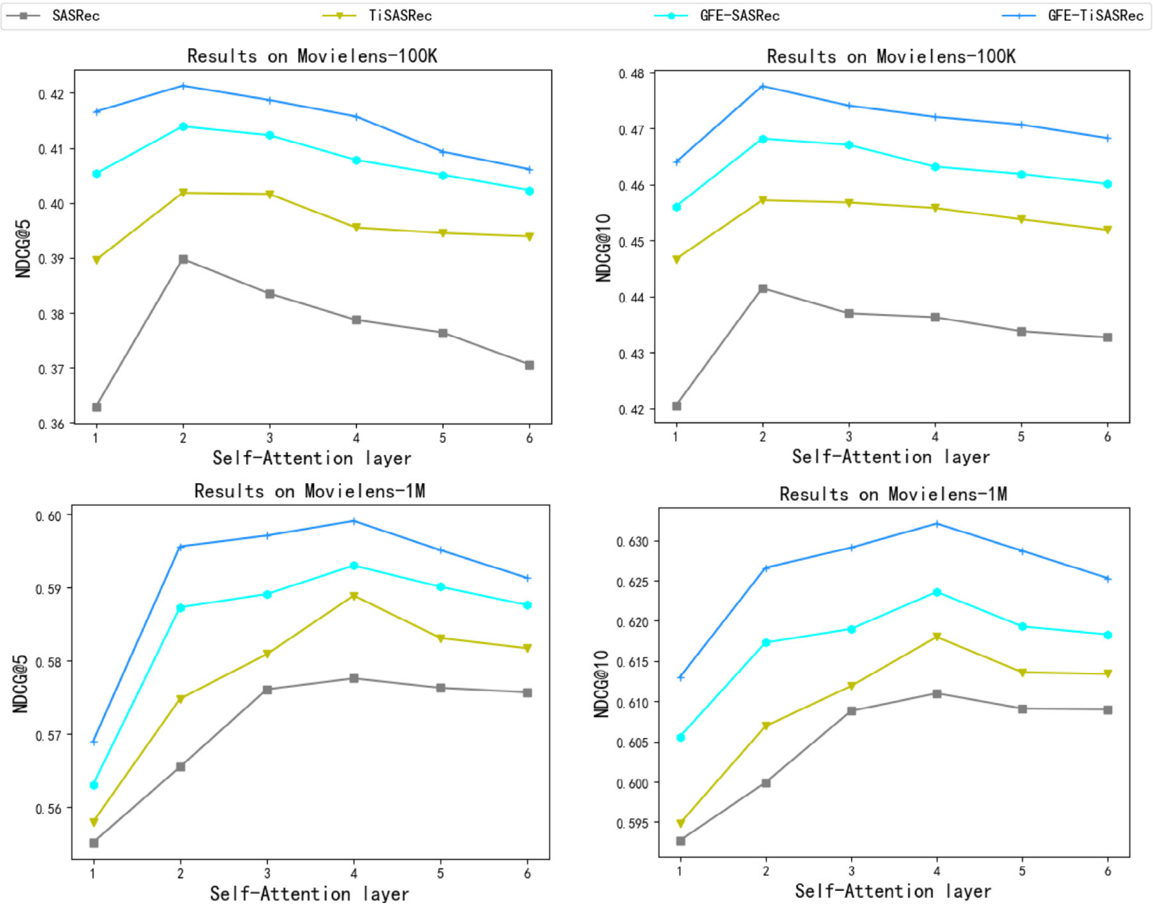

Fig. 7. The results of different self-attention layers.

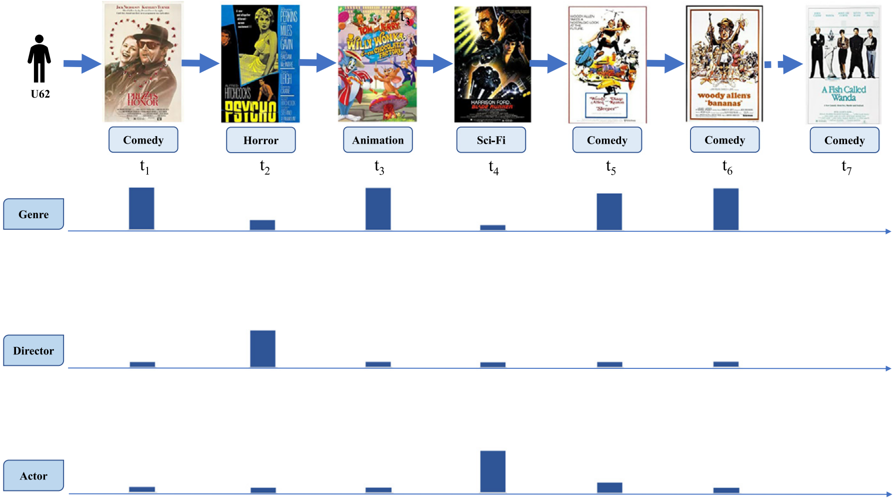

Fig. 8. The example of $u62$ for the illustration of GFE's explainability.

we propose a hierarchical self-attention mechanism to capture user's preference evolution from these knowledge paths. Based on these two strong abilities, GFE can offer better explanations from the two points of view of microcosm and macrocosm. Finally, unlike other traditional explainable sequential recommendation, GFE has the strong generalization to integrate with other pure sequential recommendation models and endow explainability to them, which has been proven in the integration of GFE and SASRec/TiSASRec. Finally, extensive experiments on three public datasets show the state-of-the-art performance and high explainability of the GFE.

In the future, we plan to extend this work in two ways. There is a common phenomenon that the KG is usually incomplete. Therefore, it is worth applying knowledge completion technique into our model, which may provide more complete and reasonable KG facts for recommender system. Besides, since the KG used in our paper is based on the item's side information, leveraging user's profiles is another promising approach to improve the performance of sequential recommendation.

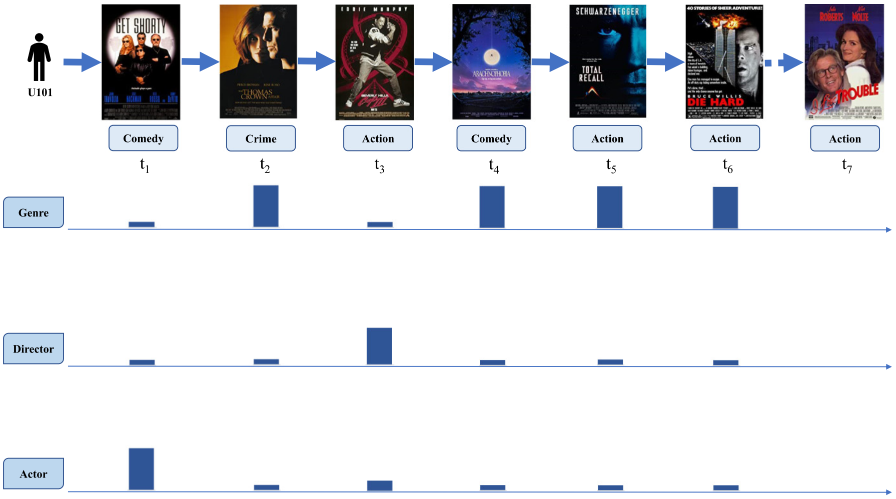

Fig. 9. The example of $u101$ for the illustration of GFE's explainability.

CRediT authorship contribution statement

Zuoxi Yang: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Shoubin Dong:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Jinlong Hu:** Supervision, Validation, Visualization, Writing - review & editing.

Declaration of competing interest

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

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61976239), Science and Technology Project in Guangzhou of China (Grant No. 201903010046) and Innovation Foundation of High-end Scientific Research Institutions in Zhongshan of China (Grant No. 2019AG031).

References

[1] S. Rendle, C. Freudenthaler, L. Schmidt-Thieme, Factorizing personalized markov chains for next-basket recommendation, in: Proceedings of the 19th International Conference on World Wide Web, 2010, pp. 811–820.

[2] G. Shani, D. Heckerman, R.I. Brafman, C. Boutilier, An MDP-based recommender system, J. Mach. Learn. Res. 6 (9) (2005).

[3] C. Cheng, H. Yang, M.R. Lyu, I. King, Where you like to go next: Successive point-of-interest recommendation, in: Twenty-Third International Joint Conference on Artificial Intelligence, 2013.

[4] J. Tang, K. Wang, Personalized top-n sequential recommendation via convolutional sequence embedding, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 565–573.

[5] A. Yan, S. Cheng, W.-C. Kang, M. Wan, J. McAuley, CosRec: 2D convolutional neural networks for sequential recommendation, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 2173–2176.

[6] B. Hidasi, A. Karatzoglou, Recurrent neural networks with top-k gains for session-based recommendations, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 843–852.

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

[8] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, J. Ma, Neural attentive session-based recommendation, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 1419–1428.

[9] W. Pei, J. Yang, Z. Sun, J. Zhang, A. Bozzon, D.M. Tax, Interacting attention-gated recurrent networks for recommendation, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 1459–1468.

[10] M. Quadrana, A. Karatzoglou, B. Hidasi, P. Cremonesi, Personalizing session-based recommendations with hierarchical recurrent neural networks, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, 2017, pp. 130–137.

[11] A. Beutel, P. Covington, S. Jain, C. Xu, J. Li, V. Gatto, E.H. Chi, Latent cross: Making use of context in recurrent recommender systems, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 46–54.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 6000–6010.

[13] W.-C. Kang, J. McAuley, Self-attentive sequential recommendation, in: 2018 IEEE International Conference on Data Mining, ICDM, IEEE, 2018, pp. 197–206.

[14] J. Li, Y. Wang, J. McAuley, Time interval aware self-attention for sequential recommendation, in: Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 322–330.

[15] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Neural Information Processing Systems, NIPS, 2013, pp. 1–9.

[16] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, pp. 1112–1119.

[17] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2181–2187.

[18] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, pp. 687–696.

[19] F. Zhang, N.J. Yuan, D. Lian, X. Xie, W.-Y. Ma, Collaborative knowledge base embedding for recommender systems, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 353–362.

[20] H. Wang, F. Zhang, X. Xie, M. Guo, DKN: Deep knowledge-aware network for news recommendation, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1835–1844.

[21] V. Bellini, V.W. Anelli, T. Di Noia, E. Di Sciascio, Auto-encoding user ratings via knowledge graphs in recommendation scenarios, in: Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems, 2017, pp. 60–66.

[22] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, M. Guo, Ripplenet: Propagating user preferences on the knowledge graph for recommender systems, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 417–426.

[23] Y. Cao, X. Wang, X. He, Z. Hu, T.-S. Chua, Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences, in: The World Wide Web Conference, 2019, pp. 151–161.

[24] X. Xin, X. He, Y. Zhang, Y. Zhang, J. Jose, Relational collaborative filtering: Modeling multiple item relations for recommendation, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 125–134.

[25] A. Gazdar, L. Hidri, A new similarity measure for collaborative filtering based recommender systems, Knowl.-Based Syst. 188 (2020) 105058.

[26] A. Salamat, X. Luo, A. Jafari, HeteroGraphRec: A heterogeneous graph-based neural networks for social recommendations, Knowl.-Based Syst. 217 (2021) 106817.

[27] Z. Zhao, X. Zhang, H. Zhou, C. Li, M. Gong, Y. Wang, HetNERec: Heterogeneous network embedding based recommendation, Knowl.-Based Syst. 204 (2020) 106218.

[28] Z. Ali, G. Qi, K. Muhammad, B. Ali, W.A. Abro, Paper recommendation based on heterogeneous network embedding, Knowl.-Based Syst. 210 (2020) 106438.

[29] K. Ji, R. Sun, W. Shu, X. Li, Next-song recommendation with temporal dynamics, Knowl.-Based Syst. 88 (2015) 134–143.

[30] L. Liu, L. Wang, T. Lian, CaSe4SR: Using category sequence graph to augment session-based recommendation, Knowl.-Based Syst. 212 (2021) 106558.

[31] C. Ma, L. Ma, Y. Zhang, J. Sun, X. Liu, M. Coates, Memory augmented graph neural networks for sequential recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, No. 04, 2020, pp. 5045–5052.

[32] R. He, J. McAuley, Fusing similarity models with Markov chains for sparse sequential recommendation, in: 2016 IEEE 16th International Conference on Data Mining (ICDM), IEEE, 2016, pp. 191–200.

[33] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, M. Guo, Multi-task feature learning for knowledge graph enhanced recommendation, in: The World Wide Web Conference, 2019, pp. 2000–2010.

[34] X. Tang, T. Wang, H. Yang, H. Song, AKUPM: Attention-enhanced knowledge-aware user preference model for recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 1891–1899.

[35] J. Huang, W.X. Zhao, H. Dou, J.-R. Wen, E.Y. Chang, Improving sequential recommendation with knowledge-enhanced memory networks, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 505–514.

[36] X. Huang, Q. Fang, S. Qian, J. Sang, Y. Li, C. Xu, Explainable interaction-driven user modeling over knowledge graph for sequential recommendation, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 548–556.

[37] P. Shaw, J. Uszkoreit, A. Vaswani, Self-attention with relative position representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 2018, pp. 464–468.

[38] R. He, J. McAuley, Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering, in: Proceedings of the 25th International Conference on World Wide Web, 2016, pp. 507–517.

[39] W.X. Zhao, G. He, K. Yang, H. Dou, J. Huang, S. Ouyang, J.-R. Wen, Kb4rec: A data set for linking knowledge bases with recommender systems, Data Intell. 1 (2) (2019) 121–136.

[40] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, in: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, 2009, pp. 452–461.

[41] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 173–182.

[42] A. Karatzoglou, X. Amatriain, L. Baltrunas, N. Oliver, Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering, in: Proceedings of the Fourth ACM Conference on Recommender Systems, 2010, pp. 79–86.

[43] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, P. Jiang, BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 1441–1450.

[44] D. Kingman, J. Ba, Adam: A method for stochastic optimization. Conference paper, in: 3rd International Conference for Learning Representations, 2015.