Profiling temporal learning interests with time-aware transformers and knowledge graph for online course recommendation

Jilei Zhou¹ · Guanran Jiang¹ · Wei Du¹· Cong Han¹

Abstract

Profiling users’ temporal learning interests is key to online course recommendation. Previous studies mainly profile users’ learning interests by aggregating their historical behaviors with simple fusing strategies, which fails to capture their temporal interest patterns underlying the sequential user behaviors. To fill the gap, we devise a recommender that incorporates time-aware Transformers and a knowledge graph to better capture users’ temporal learning interests. First, we introduce stacked Transformers to extract users’ temporal learning interests underlying users’ course enrollment sequences. In addition, we design a time-aware positional encoding module to consider the enrollment time intervals between courses. Third, we incorporate a knowledge graph to utilize the latent knowledge connections between courses. The proposed method outperforms state-of-the-art baselines for course recommendation. Furthermore, findings in the ablation study offers several insights for future research. The proposed model can be implemented in online learning platforms to increase user engagement and reduce dropout rate.

Keywords Temporal learning interests · Time-aware transformers · Knowledge graph · Course recommendation

Jilei Zhou and Guanran Jiang contributed equally to this paper.

Wei Du
ahduwei@ruc.edu.cn

¹ School of Information, Renmin University of China, Beijing 100872, People’s Republic of China
Online learning platforms such as Coursera.org, edX.org, and Udemy.com have provided low-cost opportunities for anyone to access a massive number of courses. The courses on online learning platforms see exponential growth during the COVID-19 pandemic [1]. However, the massive number of courses makes it difficult for users to choose a proper one, severely hampering both user retention rate and course completion rate on online learning platforms. In this regard, investigating an effective way of personalized course recommendation is in pressing need for online learning platforms.

Profiling users’ temporal learning interests is key to online course recommendation. Previous studies have provided evidence that individuals’ learning interests are dynamic and evolutive over time [2]. Moreover, accumulated evidence has shown that a temporal view helps to understand online learning behaviors. For example, temporal learning patterns differ between completers (i.e., learners who persist over time) and non-completers (i.e., learners who drop out) [3]. Temporal patterns extracted from students’ online learning activities also help to distinguish higher-performing students from lower-performing students [4, 5]. Therefore, it requires studying users’ interests from a temporal view and investigating online course recommendations to capture users’ temporal learning interests.

Filtering-based methods, including content-based (CB) [6, 7] and collaborative filtering (CF) [8, 9], are much-discussed in previous studies to capture users’ temporal learning interests. However, these methods profile user interests by averaging users’ historical behaviors (e.g., course enrollment, course bookmarking, etc.) with fixed weights, and the simple fusing strategy fails to capture users’ temporal learning interests underlying the sequential user behaviors. The temporal learning interests of online users can be expressed in two forms: interest drift and interest evolution. Instead of regarding user interests as constant over time, interest drift indicates users’ dynamic and diversified interests in course learning [10–12]. For example, one may be interested in courses related to blockchain given its popularity, while one’s interest may turn to epidemiology during the COVID-19 pandemic. Interest evolution indicates users’ evolving learning interests as they gain knowledge, skills, and experience over time [13, 14]. For example, one is likely to study “C++ Advanced Programming” after studying the course of “C++ Basic Programming” since the former is an advanced course of the latter. However, the aforementioned forms of temporal learning interests underlying users’ course enrollment sequences have remained uninvestigated in course recommender designs. The course enrollment sequence is a list of sequential courses that a user previously enrolled in. We seek to fill the research gap identified above by profiling temporal learning interests in users’ sequential course enrollment behaviors for online course recommendations.

Deep Recurrent Neural Networks (RNN) are widely adopted to capture temporal user interests over time by encoding historical sequential behaviors into a hidden state vector [15, 16]. Although the state vector helps to encode sequential dependency among historical enrollment behaviors, it fails to mine the temporal learning interests from the latent knowledge connections between courses. Here, latent
knowledge connections refer to high-order relationships between two courses. As shown in Fig. 1, one of the latent knowledge connections between “C++ Basic Programming” and “Introduction to Statistics” is that both of them are courses offered by the same university. Similarly, there is a high-order connection between “C++ Basic Programming” and “FinTech Foundations”, since both of them have concepts belonging to computer science field. These latent knowledge connections are helpful for extending a user’s learning interests reasonably and increasing the diversity of recommended courses. Recently, knowledge graph, a knowledge base that uses a graph-structured topology to integrate varied side information, has been widely used to incorporate domain knowledge into the recommenders [17, 18]. In our context, a knowledge graph can provide explicit and implicit knowledge connections between courses, thereby a recommender incorporating a knowledge graph can better capture and reasonably extend temporal learning interests underlying semantic relatedness among courses.

To address the aforementioned challenges, we devise a recommender that incorporates deep learning and a knowledge graph to better capture users’ temporal learning interests. The proposed method recommends the courses to a target user according to one’s sequential course enrollment behaviors. The method has three innovative designs. First, it introduces stacked Transformers to extract users’ temporal learning interests underlying their sequential enrollment behaviors. The multi-head self-attention mechanism in the Transformer module can automatically capture the dependency among courses within a sequence. Second, it augments the Transformers by encoding the time interval between the enrollment time of the first course and the enrollment time of each subsequent course in the course learning sequence. Such time-aware design captures learning interests that are dependent on time. For example, on online learning platforms, users often enroll in many diversified courses in a short period of time but only enroll in the advanced courses after finishing the basic courses. Third, it incorporates external knowledge into the sequential recommender to utilize the latent knowledge connections between courses to enhance recommendation performance. Specifically, it constructs a knowledge graph to incorporate rich meta information of courses and leverages knowledge graph embeddings to represent courses by considering their rich semantics and high-order connectivity.

We evaluate the proposed method with real-world data from XuetangX, one of the largest online learning platforms in China. The experimental results validate the
outperformance of the proposed method in generating accurate course recommenda-
tions. The results also justify the design of time-aware Transformers and knowledge
graph incorporation in improving recommendation effectiveness.

To the best of our knowledge, this research is the first attempt to profile users’
temporal learning interests from the historical behavior sequences to support course
recommendations. The temporal learning interests have been a relatively impor-
tant research question, yet profiling temporal learning interests from users’ sequen-
tial behaviors to improve course recommendation performance has remained little
discussed. To better recommend courses to the user, we propose a deep learning
method with several innovative designs to capture temporal learning interests under-
lying users’ course enrollment sequences. Our research goes further and illuminates
topics for future theoretical research based on the interesting findings of ablation
study, such as measuring the impacts of different types of domain knowledge on
users’ online learning performances and combing the sequence and timing of user
activities to understand their temporal learning interests. Our research also offers
valuable managerial implications. The proposed method can be applied to online
learning platforms to recommend proper courses to learners according to their
dynamic interests, and is expected to increase user engagement and reduce dropout
rate.

The rest of the paper is organized as follows. Section 2 reviews the previous work
related to course recommendations. The details of the proposed method are intro-
duced in Sect. 3. Section 4 provides the experimental results. Finally, Sect. 5 sum-
marizes the conclusions and discusses future implications.

2 Related works

This study aims to profile temporal learning interests for online course recommenda-
tions. Therefore, in Sect. 2.1, we review the literature on course recommendation,
followed by how our study contributes to this stream of literature. Then, in Sect. 2.2,
we discuss the literature on sequential recommendation that offers insights for mod-
eling users’ sequential behaviors to capture temporal learning interests.

2.1 Course recommendation

Course recommendation on online learning platforms is a much-discussed topic in
previous studies. Course recommendation systems profile user interests by analyzing
their historical behaviors (e.g., course enrollment or video watching) and then rec-
commend courses of users’ interests. Conventional course recommendation methods
are mainly categorized into CB approaches and CF approaches.

CB approaches recommend courses that are similar to those that a user liked in
the past. Ghauth and Abdullah [7] designed a CB recommendation system that rec-
commended not only similar learning materials that a user had viewed in the past
but also high-quality learning materials with good learners’ ratings. Recently, deep
learning has been applied to CB recommendation and achieved good performance. For example, to recommend proper learning resources to students, Shu et al. [6] proposed a CB recommendation algorithm based on convolutional neural network that captured latent factors in text information of learning sources. Xu and Zhou [19] extracted multimodal course features based on deep learning models to improve CB course recommendations.

CF approaches recommend courses that are taken by similar users. Ray and Sharma [20] developed a CF course recommendation method that leveraged course grades to measure learners’ similarities. Pang et al. [8] augmented CF course recommendation with multi-layer bucketing structures which helped to cut down the time cost of the traditional CF method. Huang et al. [9] used the CF approach to accurately predict the score of the optional course for each student by using the course score distribution of the most similar senior students. Based on the generating the predicted scores of all optional courses, the optional courses with the highest predicted scores without time conflict would be recommended to the student.

Recently, increasing research has tried to enhance the performance of conventional course recommendations on online learning platforms with domain knowledge. For example, Jing and Tang [21] proposed a course recommendation algorithm that incorporated course content and course prerequisite relations into the CF approach. Tarus et al. [22] leveraged the domain knowledge about the learners and learning recourses to calculate similarities in the CF based course recommendation system. Ibrahim et al. [23] augmented CB and CF by integrating course ontology, student ontology, and job ontology. Ontology is a hierarchically structured set of terms for describing domain knowledge and helps understand learners’ learning interests [24].

Given its advantages in leveraging domain knowledge, knowledge graph has been widely introduced to improve knowledge-aware recommenders in varied fields, for example, music recommendation [25], medicine [26], and news recommendation [27]. In recent years, a few studies in learning literature have attempted to utilize knowledge graphs to organize learning content. Aliyu et al. [28] developed a knowledge graph for university course management. Huang and Zhu [29] went further and proposed an English learning material recommendation system that combined knowledge graph with CB and CF approach. Moreover, there is an increasing interest in designing learning path recommendation system. Based on a multidimensional knowledge graph, Shi et al. [30] first found all possible learning paths and then recommended a learning path with the highest score. The score is the linear weighted sum of a feature set, such as publication time, citation count, search frequency, etc. Similarly, Chen et al. [31] designed a learning path recommendation by integrating a course knowledge graph into rule-based and machine learning based algorithms.

Standing on these recent studies, we also focus on knowledge graph-based recommendation for online learning platforms. Although several studies have introduced knowledge graph into recommendation system, they mainly profiled users’ learning interests by simply fusing users’ historical behaviors. The temporal learning interests underlying users’ behavior sequences remain uninvestigated in course recommender designs. Therefore, our study aims to augment the graph-based recommendation by profiling temporal learning interests in users’ behavior sequences for online course recommendations.
2.2 Sequential recommendation

To capture the sequential dependency among users’ historical behaviors, various methods have been proposed to make sequential recommendations. Sequential recommendation takes one’s historical sequential behaviors as input, detects behavior patterns, models sequential dependencies, and predicts the successive item that a user is likely to interact with.

The idea of early works on sequential recommendation is to combine sequential pattern mining with conventional recommendation methods. Markov Chains (MCs) are the most commonly used methods to mine sequential patterns [14–32]. For example, Rendle et al. [32] proposed a factorizing personalized Markov Chains (FPMC) model for next basket recommendation, which combined first-order MCs and matrix factorization (MF) to model sequential behaviors on e-commerce platforms. Based on the FPMC method, He and McAuley [33] proposed a method by fusing similarity-based methods with high-order MCs to make personalized sequential recommendations. A major problem of these methods is that they still rely heavily on conventional recommendation methods that simply fuse historical behaviors.

Since RNN and its variants (e.g., long short-term memory (LSTM) and gated recurrent unit (GRU)) demonstrate their effectiveness in processing sequence data, RNN-based models have attracted increasing interests in sequential recommendations [16–35]. For example, based on RNN, Yu et al. [34] proposed a next basket recommendation model called DREAM which captured global sequential features among baskets. Donkers et al. [16] proposed a sequential recommendation method that introduced GRU to model the temporal dynamics of consumption sequences. Similarly, Wei et al. [35] proposed a sequential recommender system called LANCR, where LSTM captured users’ long-term interests. However, these RNN-based recommendations are left-to-right unidirectional models where each item can only encode the information from previous items, which assume a rigidly ordered sequence over data. It is not always true for user behaviors in real-world applications [36, 37]. That is, RNN-based recommendations are not sufficient to learn optimal representations for the temporal interests underlying users’ sequential behaviors.

To overcome these obstacles, a new model named Transformer has achieved admirable success in the recommendation systems [38, 39]. Unlike RNN and its variants, Transformer allows the model to access any part of the history regardless of distance, making it potentially more suitable for grasping recurring patterns with long-term dependencies [40]. For example, based on the powerful Transformer model, Chen et al. [38] captured the sequential signals underlying users’ behavior sequences for recommendation on e-commerce platforms. Sun et al. [39] developed a sequential recommendation model called BERT4Rec that employed the Bidirectional Encoder Representations from Transformer (BERT) to model user behavior sequences.

In summary, previous course recommendation research adopt a simple strategy of fusing historical behaviors while failing to consider the sequential pattern of learning behaviors. Inspired by its great success for sequential recommendation, we try to
use Transformer to profile temporal learning interests underlying users’ sequential behaviors. Furthermore, we propose to augment the Transformer model with knowledge-aware embedding, time-aware positional encoding, and stacking Transformer structures.

3 Proposed method

This section presents the proposed model in detail. As shown in Fig. 2, the proposed model takes a user’s course enrollment sequence as input, and outputs the recommendation probability of a candidate course \( v \) to a target user \( u \).

First, a knowledge graph incorporating the rich course metadata and relations between courses is constructed to support the course recommendation. Knowledge-aware embedding layer embeds courses into low-dimensional vectors through knowledge graph embedding, which helps to capture semantic relatedness among courses. Second, we profile the user’s temporal learning interests from his/her course enrollment sequence, i.e., \( \{v_1, v_2, \ldots, v_j\} \). We propose innovative Time-aware Transformers to profile temporal learning interests by considering the enrollment time of historical courses and the different levels of attention on historical courses in the recommendation. Lastly, the recommendation probability of a candidate course is predicted with a Deep Neural Networks (DNN) layer by concatenating the vector representations of the user and a candidate course.
3.1 Knowledge-aware embedding layer

The course knowledge graph is a graph with a large number of entities (i.e., nodes) and relations (i.e., edges). We design a knowledge graph schema, which defines the knowledge scope, to cover rich knowledge relating to courses for online course recommendations (see the schema in Fig. 3). As we can see, entities (i.e., course, teacher, school, concept, and field) and relations between them are fused into the course knowledge graph. Notably, the course knowledge graph can be extended to include more entities and relations with more information sources.

There are massive facts in the course knowledge graph (see the instances in Fig. 3). For example, course “C++ basic programming” is connected to course “C++ advanced programming” because the former is a prerequisite course of the latter. Moreover, course “C++ basic programming” is similar to course “C++ advanced programming” because several concepts of these two courses, such as “C++”, “class”, and “binary tree”, are hyponyms of “computer science” field. Each fact can be represented as an entity-relation-entity triple \((h, r, t)\), where \(h\), \(r\), and \(t\) represent the head, the relation, and the tail of a triple, respectively. Formally, we define the course knowledge graph as

\[
G = \{h, r, t | h, t \in \mathcal{E}, r \in \mathcal{R}\} 
\]

where \(\mathcal{E}\) is the entity set and \(\mathcal{R}\) is the relation set in \(G\).

Knowledge graph embedding is an effective way to parameterize entities and relations as vector representations, while preserving the graph structure. Among various knowledge graph embedding techniques, translation-based models, including TransE [41], TransH [42], TransR [43], and TransD [44], are popular ones given their high scalability to large real-world datasets. The basic idea of the translation-based models is that knowledge graph embedding can be learned by considering the relation \(r\) from the head entity \(h\) to the tail entity \(t\) as distance or dissimilarity. TransE as the first translation-based model has gained wide popularity due to its high efficiency. However, it fails to deal with the one-to-many/many-to-one/many-to-many relations in real-world knowledge graphs. TransH, TransR, and TransD are successively proposed to overcome the flaws of TransE. Among which, TransH...
and TransR mainly focus on modeling various relations while neglecting the difference between entities and relations. To deal with this problem, TransD is proposed to consider the diversity of both relations and entities by mapping them into separate paces. Moreover, TransD achieves superior performance compared with state-of-the-art translation-based models. Therefore, we adopt TransD in our study as the knowledge graph embedding technique. We can learn the representation of course entities with TransD. We use vector \( \mathbf{c} \) to represent the candidate course and a sequence of vectors \( \{ \mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_j \} \) to denote the knowledge graph embeddings of courses historically enrolled by the user \( u \).

### 3.2 Time-aware transformers

We design time-aware transformers to mine the temporal learning interests from the course enrollment sequences. Within the time-aware transformers, a time-aware positional encoding module is proposed to capture the orders and the enrollment time of courses in course learning sequences. Stacked Transformer layers instead of a single Transformer layer are introduced to form a deep network to learn more complex item transition patterns [39–45]. We introduce the time-aware Transformers in detail by following the calculation process.

The conventional positional encoding used in Transformers injects information about the relative or absolute position/order of each item in the sequence [40–46], but fails to capture the course enrollment behaviors within different time intervals. To better capture course learning interests that are dependent on time, we design a time-aware positional encoding module to encode not only the order but also the enrollment time of each course in a course enrollment sequence. For each position index \( k(1 \leq k \leq j) \), we calculate its time-aware positional value as

\[
TPos(v_k) = k + t(v_k)
\]

where \( t(v_k) \) is the time interval between the enrollment time of the course \( v_k \) and the enrollment time of the first course \( v_1 \) in the sequence.

To encode the time-aware positional value \( TPos(v_k) \) into the corresponding input course embedding \( \mathbf{c}_k \), we extend \( TPos(v_k) \) into a \( n \)-dimensional positional embedding \( \mathbf{p}_k \) by following Vaswani et al. [40]’s study [40]. For a given dimension \( i(1 \leq i \leq n) \) within \( \mathbf{p}_k \), we calculate the value of each dimension as follows:

\[
p_k(i) = \begin{cases} 
sin \left( \frac{TPos(v_k)}{10000 \frac{\pi}{2}} \right), & \text{if } i \text{ is even} \\
\cos \left( \frac{TPos(v_k)}{10000 \frac{\pi}{2}} \right), & \text{if } i \text{ is odd}
\end{cases}
\]

(2)

For a given course \( v_k \), its input representation \( \mathbf{h}^0_k \) is constructed by summing the knowledge graph embedding \( \mathbf{c}_k \) and the time-aware positional embedding \( \mathbf{p}_k \):

\[
\mathbf{h}^0_k = \mathbf{c}_k + \mathbf{p}_k
\]

(3)
Given the embedding matrices of all courses $\mathbf{H} = [\mathbf{h}_1^0, \mathbf{h}_2^0, \ldots, \mathbf{h}_j^0]$, we introduce the stacked Transformer layers to capture the dependency among courses within a sequence. Each Transformer layer contains two sub-layers, a Multi-Head Self-Attention Layer and a Position-wise Feed-Forward Layer. The multi-head self-attention layer allows the Transformer to jointly attend to information from different representation subspaces at different positions. The position-wise feed-forward layer endows the Transformer with nonlinearity and interactions between different dimensions. The multi-head attention is computed as

$$\text{MultiHead}(\mathbf{H}) = \text{Concat}(\text{head}_1; \text{head}_2; \ldots; \text{head}_h) \mathbf{W}^O$$  \hspace{1cm} (4)

where $h$ is the number of heads, and $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V$ are the projection matrices for each head. The Attention function is scaled dot-product attention:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}$$  \hspace{1cm} (6)

where $\mathbf{Q}$ represents the queries, $\mathbf{K}$ the keys and $\mathbf{V}$ the values. The $\sqrt{d}$ is introduced to produce a softer attention distribution for avoiding extremely small gradients.

Then, we apply a position-wise feed-forward layer to the outputs of the multi-head self-attention layer, separately and identically at each position. It consists of two linear transformations with a ReLU activation in between:

$$\text{FFN}(x) = \max(0, (xW_1 + b_1)W_2 + b_2)$$  \hspace{1cm} (7)

where $W_1$, $W_2$, $b_1$, and $b_2$ are learnable parameters and shared across all positions.

After $L$ layers, we get the final output $\mathbf{H}^L = [\mathbf{h}_1^L, \mathbf{h}_2^L, \ldots, \mathbf{h}_j^L]$ for all courses in the input sequence.

### 3.3 Recommend probability calculation

Given the outputs of the time-aware Transformers $\mathbf{H}^L = [\mathbf{h}_1^L, \mathbf{h}_2^L, \ldots, \mathbf{h}_j^L]$ and the knowledge graph embedding of the candidate course $\mathbf{c}$, we first concatenate the two embeddings as the input, then the probability of the user enrolling in the candidate course $v$ is predicted by a DNN.

DNN is a neural network that has an input layer, an output layer and at least one hidden layer in between. In our study, the input layer of DNN is the concatenation of vector representations of the user and a candidate course. Then, the DNN adds several hidden layers to capture more complex relationships from input information. The non-linear activation function between the input layer and the hidden layer is $\psi(\mathbf{c}) = \text{ReLU}(\sum w_ix_i + b)$, where $w_i$ is the coefficient of linear relation, and $b$ is the bias. Finally, DNN maps the output of the hidden layer $\psi(\mathbf{c})$ through a softmax layer to a probability distribution. The probability of a candidate course $v$ is given by
where $V^T$ is the matrix of weights of the softmax layer. $h(\cdot)$ is the softmax function given by $h(v) = \frac{e^v}{\sum_p e^p}$ where $p$ is the total number of candidate courses.

### 4 Experimental design

#### 4.1 Datasets

We collected real-world data from XuetangX,\(^1\) one of the largest online learning platforms in China. The dataset includes enrollment records of 199,199 users and 706 courses from June 23th, 2015 to November 13th, 2019. Most users enrolled in very few courses, e.g., 75% of users enrolled in less than five courses. Therefore, we extracted 937 active users with more than 20 enrollment records in the dataset to evaluate our model. A course knowledge graph was constructed for the 706 courses, and the statistics are listed in Table 1. The course knowledge graph consists of 706 courses, 1737 teachers, 112,844 knowledge concepts, and multiple corresponding types of relationships.

#### 4.2 Evaluation metrics

To evaluate the performance of the proposed model, we employed a variety of evaluation metrics, including recall and normalized discounted cumulated grain (nDCG). Recall@\(k\) measures the percentage of the correctly predicted items set $R$ in the ground truth set $S$ in the top-$k$ recommendation list:

$$Recall@k = \frac{1}{|u|} \sum_{u \in USER} \frac{|R \cap S|}{S}$$  \hspace{1cm} (9)

### Table 1  Statistics of the course knowledge graph

| Entities          | Relations                      |
|-------------------|--------------------------------|
| #Course: 706      | #Prerequisite Course: 1027     |
| #Teacher: 1737    | #Teach: 2349                   |
| #School: 208      | #List: 705                     |
| #Concept: 112,844 | #Has Concept: 167,751          |
| #Field: 1719      | #Hyponym Of: 114.563           |

\(^1\) http://www.xuetangx.com

\[RecProb(c, v) = h(\psi(c)V^T)\]  \hspace{1cm} (8)
nDCG@k is a metric that further considers the position of recommended items in the top-k recommendation list, and is widely used to measure ranking quality. nDCG@k is calculated using the following formula:

\[
nDCG@k = \frac{DCG@k}{IDCG@k}
\]

(10)

where DCG@k is calculated as follows:

\[
DCG@k = \sum_{i=1}^{\lfloor REL \rfloor} \frac{rel_i}{log_2(i + 1)},
\]

(11)

and IDCG@k is the maximum DCG@k value under ideal conditions:

\[
IDCG@k = \sum_{i=1}^{\lfloor REL \rfloor} \frac{2^{rel_i} - 1}{log_2(i + 1)},
\]

(12)

where \(\lfloor REL \rfloor\) represents the best ranking list where the items with true label of 1 are recommended ahead. Higher nDCG@k values indicate that more correctly recommended items are ranked on top.

### 4.3 Baseline methods

We evaluated our model, abbreviated as KT_Transformers, by comparing it with several baseline models. Conventional recommendation models that do not consider entity embeddings learned from knowledge graph, i.e., POP, ItemCF, UserCF, BPR, and FPMC, are selected as baselines. Further, recommendation models that consider knowledge graph but neglect sequence features, i.e., DNN, ATT4REC, and DKN, are also compared. Last, recommendation models that incorporate both knowledge graph and sequence information, i.e., LSTM4REC and GRU4REC, are included as baselines.

**POP** POP calculates the popularity of items based on the user-item interactions and recommends the top popular items.

**ItemCF** In the ItemCF method, item similarities are calculated by using a user-item matrix which are used to predict ratings for user-item pairs.

**UserCF** Unlike the ItemCF method, UserCF uses user similarities calculated by the user-item matrix to identify the nearest neighbors. A user’s ratings on corresponding items are calculated by the nearest neighbors’ rating on items.

**BPR** [47] Bayesian personalized ranking (BPR) uses matrix factorization for the non-sequential recommendation.

**FPMC** [32] This model combines matrix factorization machine and MCs for next item recommendations by capturing time information and long-term user preferences.

**DNN** Deep Neural Networks (DNN) use a feed-forward neural network to learn high-order features from entity embeddings extracted from knowledge graph.
ATT4REC [48] This model incorporates knowledge graph embeddings and uses the self-attention mechanism to infer the relationships between items from the user’s historical interactions, and considers both short-term and long-term preferences.

DKN [49] Deep knowledge-aware network (DKN) is a classic knowledge-aware recommendation framework. It adopts a multi-channel and word–entity-aligned knowledge-aware CNN and introduces an attention module to dynamically aggregate a user’s history.

LSTM4REC LSTM4REC includes knowledge graph embedding and uses LSTM to model sequential data, which solves the long-term dependency problem through the forgetting gate.

GRU4REC [15] It adopts entity embeddings learned from knowledge graph as the input and uses the gated recurrent unit (GRU) to learn the historical session records of the target user.

4.4 Model selection and parameter setting

To learn the knowledge graph embedding, we adopted TransD as a knowledge graph embedding technique. TransD considers the diversity of both entities and relations in the knowledge graph and achieves superior performance compared with other translation-based models. Moreover, TransD has better efficiency on large-scale graphs because it has fewer parameters and no matrix–vector multiplication operations. To train the proposed model, we randomly selected 70% of users as the training set and 30% of users as the test set. As mentioned before, we proposed a deep network by stacking Transformer layers to learn more complex item transition patterns. However, the network becomes more difficult to train as it goes deeper. To enable deep training and avoid overfitting, we set the number of Transformer layers as two. Following the negative sampling strategy used in Wang et al. [18], for each positive sample, we randomly chose three or four negative samples to enlarge the dataset. The embedding size was set as 16, 32, 64, 128, and 256. The dimension of 64 achieved the best performance. After comparing Adam and SGD, we chose SGD as the optimizer because the optimization effect of SGD was more stable. To balance the learning speed and effect, the learning rate was set as 0.00001 for all models by comparing the performance of different values in {0.0001, 0.00001, 0.000001}. We set the dropout rate as 0.5 for all models. The weight coefficients and other parameters were adjusted automatically in the process of model optimization.

5 Results and discussion

5.1 Recommendation performance evaluation and comparison with baselines

Table 2 shows the recommendation performance of the proposed model (i.e., KT_Transformers) and baseline methods in terms of nDCG@k and Recall@k. The
Overall, KT_Transformers outperforms baseline methods significantly, which demonstrates the capability of KT_Transformers in generating precise recommendations.

Sequential recommendation helps improve recommendation effectiveness. In general, sequential recommendations, including FPMC, GRU4Rec, LSTM4Rec, ATT4Rec, and KT_Transformers, perform better than conventional recommenders such as POP, BPR, and ItemCF. However, the UserCF outperforms FPMC by 20.4% on NDCG@10 and 20.0% on Recall@10. The results indicate that profiling users’ temporal learning interests underlying users’ behavior sequences enhances online course recommendations.

Deep learning models have superiority in modeling users’ behavior sequences compared with MCs and therefore improve recommendation effectiveness. DNN outperforms FPMC by 23.9% on nDCG@10 and 42.9% on Recall@10. GRU4Rec outperforms FPMC by 45.9% on nDCG@10 and 41.7% on Recall@10. LSTM4Rec outperforms FPMC by 48.8% on nDCG@10 and 41.7% on Recall@10. ATT4Rec outperforms FPMC by 157.6% on nDCG@10 and 57.4% on Recall@10. KT_Transformers outperforms FPMC by 64.8% on nDCG@10 and 61.6% on Recall@10.

The design of the attention mechanism was also validated. ATT4Rec with an attention mechanism performs best among all baselines. KT_Transformers with advanced multi-head attention mechanisms outperforms ATT4Rec by 16.8% on nDCG@10 and 16.4% on Recall@10. That means attention mechanisms can help to learn the dependency among items in users’ behavior sequences.

Overall, the experimental results demonstrate the efficacy of the design of our method, i.e., sequential recommendation, the use of deep learning, and the attention mechanism.

### Table 2 Performance comparison of baselines

| Models       | nDCG@10  | Recall@10 | nDCG@20  | Recall@20 |
|--------------|----------|-----------|----------|-----------|
| KT_Transformers | 0.244    | 0.073     | 0.277    | 0.128     |
| POP          | 0.009 (−96.3%) | 0.006 (−91.8%) | 0.024 (−91.3%) | 0.014 (−89.1%) |
| BPR          | 0.042 (−82.8%) | 0.014 (−80.8%) | 0.062 (−77.6%) | 0.029 (−77.3%) |
| ItemCF       | 0.055 (−77.5%) | 0.019 (−74%) | 0.076 (−72.6%) | 0.032 (−75%) |
| UserCF       | 0.108 (−55.7%) | 0.035 (−52.1%) | 0.132 (−52.3%) | 0.057 (−55.5%) |
| FPMC         | 0.086 (−64.8%) | 0.028 (−61.6%) | 0.093 (−66.4%) | 0.048 (−62.5%) |
| DNN          | 0.113 (−53.7%) | 0.049 (−32.9%) | 0.145 (−47.7%) | 0.087 (−32%) |
| ATT4Rec      | 0.203 (−16.8%) | 0.061 (−16.4%) | 0.228 (−17.7%) | 0.103 (−19.5%) |
| DKN          | 0.045 (−81.6%) | 0.018 (−75.3%) | 0.061 (−78%) | 0.033 (−74.2%) |
| GRU4Rec      | 0.159 (−34.8%) | 0.048 (−34.2%) | 0.179 (−35.4%) | 0.082 (−35.9%) |
| LSTM4Rec     | 0.168 (−31.1%) | 0.05 (−31.5%) | 0.197 (−28.9%) | 0.09 (−29.7%) |
5.2 The role of knowledge graph

One innovative design of our model is that we introduced knowledge graph embeddings into the sequential recommender to leverage the latent knowledge connections among courses. To evaluate the role of the knowledge graph, we compared KT_Transformers with T_Transformers where knowledge graph embeddings were replaced with randomly initialized embeddings. That is, KT_Transformers and T_Transformers share the same model structures but different values of knowledge graph embedding. Figure 4 shows that a knowledge graph helps improve recommendation effectiveness. With the course knowledge graph embeddings as input, nDCG@10 increases by 18% and nDCG@10 increases by 33%.

Given the importance of the course knowledge graph, we go further and investigate prescriptive insights extracted from diversification of domain knowledge. There is a wealthy of domain knowledge available on the online learning platforms that may be relevant to the learning resources (i.e., courses in our case), such as teachers, research institutions that list courses, course field, course concepts, and prerequisite learning recourses. However, it remains unclear how different types of domain knowledge drive user decisions on choosing...
learning resources. The results from the ablation experiments might provide several insights regarding this question. Specifically, we deleted one type of entity in the course knowledge in turn and updated the knowledge graph embeddings. Figure 5 shows that the model lacking the “field” entity performs worst and induces a 25% loss on NDCG@10 and 21% loss on recall@10 compared to the model with the intact knowledge graph. Compared to the “field” entity, models lacking other types of entities experience slight performance degradation, especially the “teacher” entity. That is, in the context of online learning, users’ preferences on specific fields matter more than their preferences on other types of domain knowledge. Moreover, users’ preferences on specific teachers are the least important factors in their choice decisions. In the context of offline learning, previous studies have highlighted the important roles of teacher characteristics on student outcomes [50, 51]. In view of this, our findings illuminate topics for future theoretical online learning research, such as measuring the impacts of teacher characteristics on users’ online learning performances.

5.3 The role of time-aware positional encoding

Another innovative design of our model is designing a time-aware positional encoding module to capture temporal interest patterns related to the orders and the enrollment time of courses in course learning sequences. To demonstrate the added value of considering temporal patterns of user activity when recommending courses, we compared performances of three models: NoPOS, POS, and Time + POS. For courses in a course enrollment sequence, NoPOS did not consider any position information; POS only considered the order information; and Time + POS considered both the order and the enrollment time.

The comparison in Fig. 6 shows that the Time + POS performs best among all three models. The result indicates that temporal patterns including the order and the timing of user enrollment behaviors help to course recommendation. A possible explanation is that the order and the timing of user enrollment behaviors capture two important kinds of learning interests: interest drift and interest evolution [12–14]. On the one hand, users often enroll in many courses in a short period of time, which
usually implies users’ interest drift. The time interval between the enrollment time of the first course and the enrollment time of each course in the course learning sequence helps to capture the pattern. On the other hand, users will enroll in the advanced courses after finishing the basic courses, which usually reflects users’ interest evolution as they gain knowledge. It can be captured by the order of the enrollment behaviors. Furthermore, as we mentioned before, previous studies in other fields has demonstrated the effectiveness of considering the order of user activities [15, 16]. However, it seems to be not sufficient in the context of online learning. As we can see, the \textit{POS} yields worse performance than the \textit{Time + POS}. This implies the importance of considering both the sequence and the timing of learning activities when understanding users’ temporal learning interests in future research.

5.4 A summary of experimental results

Based on real-world data collected from one online learning platform in China, our experimental results show that the proposed framework can improve the effectiveness of course recommendation. When $k$ varies from 10 to 20, the values of nDCG@$k$ are 25.5% and 27.7%, and the values of Recall@$k$ are 7.3% and 12.8%. In this regard, users’ temporal learning interests underlying the order, the timing, and the semantics of sequential enrollment behaviors are three critical determinants of accurate course recommendation. Profiling temporal learning interests with time-aware stacked Transformers and knowledge graph can not only improve the course recommendation but also explore what factors affect learners’ choices on online learning platforms. Specifically, our experiments compare prediction performance of various designs respectively. The results indicate that different types of domain knowledge have different effects on users’ enrollment decision. Besides, we demonstrate the added value of considering the order and the timing information respectively when recommending courses.

6 Conclusion and discussion

This study focuses on the course recommendation problem in booming online learning platforms. We propose a deep learning recommender that incorporates time-aware Transformers and a knowledge graph to profile users’ temporal learning interests in users’ course enrollment sequences. First, we design stacked Transformers to extract users’ temporal learning interests underlying their sequential learning behaviors. Second, we develop a time-aware positional encoding module which extracts users’ temporal learning interests from the enrollment time of courses. Third, we incorporate a knowledge graph to capture users’ temporal learning interests behind the latent knowledge connections among courses. The experimental results on a real dataset show the outperformance of our model and yield several interesting findings. First, the course knowledge graph that integrates rich course information helps improve recommendation effectiveness. Second, the roles of different types of knowledge in the course knowledge graph in improving recommendation
performance vary. Third, encoding the time information of each course in the course learning sequence helps capture time-aware interest patterns. These finding illuminate topics for future theoretical online learning research.

The theoretical implications are listed as follows. On the one hand, it contributes to the stream of literature that specifically seeks to profile users’ temporal learning interests for online course recommendation. Whereas studies in this vein have taken simple fusing strategy (i.e., CB and CF approaches) to average users’ historical behaviors, our work precisely captures temporal learning interests underlying the order, the timing, and the semantics of sequential enrollment behaviors. To the best of our knowledge, this work is among the first to model interest drift and interest evolution underlying learner’ sequential enrollment behaviors by using time-aware stacked Transformers. On the other hand, it links the growing literature on knowledge graph with course recommendation, thereby advancing our understanding of the latent knowledge connections between courses and the effect of different domain knowledge on users’ enrollment behaviors.

This study also offers valuable managerial implications for online learning platforms in course recommendations. With the increase of courses on online learning platforms, it is more difficult for users to choose a proper one, which hampers both user retention rate and course completion rate. The proposed model can be implemented on online learning platforms, such as Coursera.org and edX.org, and the design can help alleviate users’ search costs from large course databases and facilitate the matching between users and courses. Besides, our study compares the performance of various designs respectively; an understanding of such results is a prerequisite for answering how to propose a course recommendation for students. We find, for example, course fields matter more than teachers to profile users’ learning preferences. This finding can inform that features related to course field may need more attention than features related to teacher when designing course recommendation in the context of online learning. We also find that both the order and the timing of user enrollment behaviors help enhance course recommendation, meaning that the order and the timing are two important determinants of accurate course recommendation. Furthermore, the design of the course knowledge graph can also be leveraged for other course analyses, such as learning performance prediction, course concept extraction, learning path analysis, etc.

This research has a number of limitations that can lead to future research. First, our study is based on a public data from one online learning platform. It will be useful to generalize our study to other types of online learning platforms, as well as to general learning scenarios. Second, our data includes only active users with more than 20 enrollment records. Since inactive users’ action sequences are usually sparse, the effectiveness of the proposed recommender may be unclear for inactive users. Based on Lu et al.’s study [52], one of the potential solutions is augmenting the sequential recommendation by retrieving users with similar preferences to enhance collaborative learning across users and thus conquering sparsity.

Our work also merits future research in the following aspects. First, in addition to enrollment behaviors, future studies could boost recommendation performance by considering other user behaviors, such as video watching behaviors and questioning behaviors. Second, the value of knowledge graph could be further explored.
For example, future research could extend the knowledge graph embedding used in this paper to learn richer content information. Third, recommendation explainability could be further explored by considering the explicit path information in the course knowledge graph.

Acknowledgements This work is partially supported by the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (Grant No. 21XNLG07), National Natural Science Foundation of China (Grant No. 71901208), and Ministry of Education, Science and Technology Development Center (Grant No. 2019J01010).

Declarations

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

References

1. Impey, C. (2020). Massive online open courses see exponential growth during COVID-19 pandemic. Retrieved from https://theconversation.com/massive-online-open-courses-see-exponential-growth-during-covid-19-pandemic-141859
2. Boroujeni, M. S., & Dillenbourg, P. (2019). Discovery and temporal analysis of MOOC study patterns. Journal of Learning Analytics, 6(1), 16–33.
3. Rizvi, S., Rienties, B., & Rogaten, J. (2018). Temporal dynamics of MOOC learning trajectories. Paper presented at the proceedings of the 1st international conference on data science, E-learning and information systems.
4. Mirzaei, M., & Sahebi, S. (2019). Modeling students’ behavior using sequential patterns to predict their performance. Paper presented at the International Conference on Artificial Intelligence in Education, Beijing, China.
5. Perera, D., Kay, J., Koprinska, I., Yacef, K., & Zaïane, O. R. (2008). Clustering and sequential pattern mining of online collaborative learning data. IEEE Transactions on Knowledge and Data Engineering, 21(6), 759–772.
6. Shu, J., Shen, X., Liu, H., Yi, B., & Zhang, Z. (2018). A content-based recommendation algorithm for learning resources. Multimedia Systems, 24(2), 163–173.
7. Ghauth, K. I., & Abdullah, N. A. (2010). Learning materials recommendation using good learners’ ratings and content-based filtering. Educational technology research and development, 58(6), 711–727.
8. Pang, Y., Jin, Y., Zhang, Y., & Zhu, T. (2017). Collaborative filtering recommendation for MOOC application. Computer Applications in Engineering Education, 25(1), 120–128.
9. Huang, L., Wang, C.-D., Chao, H.-Y., Lai, J.-H., & Philip, S. Y. (2019). A score prediction approach for optional course recommendation via cross-user-domain collaborative filtering. IEEE Access, 7, 19550–19563.
10. Widmer, G., & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. Machine learning, 23(1), 69–101.
11. Yin, H., Zhou, X., Cui, B., Wang, H., Zheng, K., & Nguyen, Q. V. H. (2016). Adapting to user interest drift for poi recommendation. IEEE Transactions on Knowledge and Data Engineering, 28(10), 2566–2581.
12. Sun, B., & Dong, L. (2017). Dynamic model adaptive to user interest drift based on cluster and nearest neighbors. IEEE Access, 5, 1682–1691.
13. Yuan, Z.-M., Huang, C., Sun, X.-Y., Li, X.-X., & Xu, D.-R. (2015). A microblog recommendation algorithm based on social tagging and a temporal interest evolution model. Frontiers of Information Technology & Electronic Engineering, 16(7), 532–540.
14. McAuley, J. J., & Leskovec, J. (2013). From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. Paper presented at the Proceedings of the 22nd international conference on World Wide Web, Rio de Janeiro, Brazil.
15. Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939.

16. Donkers, T., Loepp, B., & Ziegler, J. (2017). Sequential user-based recurrent neural network recommendations. Paper presented at the proceedings of the 7th ACM conference on recommender systems, New York, USA.

17. Huang, J., Zhao, W. X., Dou, H., Wen, J.-R., & Chang, E. Y. (2018). Improving sequential recommendation with knowledge-enhanced memory networks. Paper presented at the proceedings of the 41st international ACM SIGIR conference on research & development in information retrieval, New York, USA.

18. Wang, X., Wang, D., Xu, C., He, X., Cao, Y., & Chua, T.-S. (2019). Explainable reasoning over knowledge graphs for recommendation. Paper presented at the proceedings of the 33rd AAAI conference on artificial intelligence, Hawaii, USA.

19. Xu, W., & Zhou, Y. (2020). Course video recommendation with multimodal information in online learning platforms: A deep learning framework. British Journal of Educational Technology, 51(5), 1734–1747.

20. Ray, S., & Sharma, A. (2011). A collaborative filtering based approach for recommending elective courses. Paper presented at the proceedings of the 5th international conference on information intelligence, systems, technology and management, Gurgaon, India.

21. Jing, X., & Tang, J. (2017). Guess you like: course recommendation in MOOCs. Paper presented at the proceedings of the international conference on web intelligence.

22. Tarus, J. K., Niu, Z., & Yousif, A. (2017). A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. Future Generation Computer Systems, 72, 37–48.

23. Ibrahim, M. E., Yang, Y., Ndzi, D. L., Yang, G., & Al-Maliki, M. (2018). Ontology-based personalized course recommendation framework. IEEE Access, 7, 5180–5199.

24. Chen, C. M. (2009). Ontology-based concept map for planning a personalised learning path. British Journal of Educational Technology, 40(6), 1028–1058.

25. Oramas, S., Ostuni, V. C., Noia, T. D., Serra, X., & Sciascio, E. D. (2016). Sound and music recommendation with knowledge graphs. ACM Transactions on Intelligent Systems and Technology (TIST), 8(2), 1–21.

26. Wang, M., Liu, M., Liu, J., Wang, S., Long, G., & Qian, B. (2017). Safe medicine recommendation via medical knowledge graph embedding. arXiv preprint arXiv:1710.05980.

27. Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., & Guo, M. (2018). Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. Paper presented at the proceedings of the 27th ACM international conference on information and knowledge management (CIKM), Turin, Italy.

28. Aliyu, I., Kana, A., & Aliyu, S. (2020). Development of knowledge graph for university courses management. International Journal of Education and Management Engineering, 10(2), 1.

29. Huang, Y., & Zhu, J. (2021). A personalized English learning material recommendation system based on knowledge graph. International Journal of Emerging Technologies in Learning, 16(11), 160–173.

30. Shi, D., Wang, T., Xing, H., & Xu, H. (2020). A learning path recommendation model based on a multidimensional knowledge graph framework for e-learning. Knowledge-Based Systems, 195, 105618.

31. Chen, H., Yin, C., Fan, X., Qiao, L., Rong, W., & Zhang, X. (2021). Learning path recommendation for MOOC platforms based on a knowledge graph. Paper presented at the international conference on knowledge science, engineering and management.

32. Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L. (2010). Factorizing personalized markov chains for next-basket recommendation. Paper presented at the Proceedings of the 19th international conference on world wide web, Raleigh, USA.

33. He, R., & McAuley, J. (2016). Fusing similarity models with markov chains for sparse sequential recommendation. Paper presented at the proceedings of the 16th IEEE international conference on data mining (ICDM), Barcelona, Spain.

34. Yu, F., Liu, Q., Wu, S., Wang, L., & Tan, T. (2016). A dynamic recurrent model for next basket recommendation. Paper presented at the proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval, New York, USA.
35. Wei, X., Zuo, X., & Yang, B. (2019). *Sequential recommendation based on long-term and short-term user behavior with self-attention*. Paper presented at the Proceedings of the 12th international conference on knowledge science, engineering and management (KSEM), Athens, Greece.

36. Wang, S., Hu, L., Cao, L., Huang, X., Lian, D., & Liu, W. (2018). *Attention-based transactional context embedding for next-item recommendation*. Paper presented at the proceedings of the 32nd AAAI conference on artificial intelligence, Louisiana, USA.

37. Hu, L., Cao, L., Wang, S., Xu, G., Cao, J., & Gu, Z. (2017). *Diversifying personalized recommendation with user-session context*. Paper presented at the proceedings of the 26th international joint conference on artificial intelligence (IJCAI), Melbourne, Australia.

38. Chen, Q., Zhao, H., Li, W., Huang, P., & Ou, W. (2019). *Behavior sequence transformer for e-commerce recommendation in alibaba*. Paper presented at the proceedings of the 1st International workshop on deep learning practice for high-dimensional sparse data, New York, USA.

39. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., & Jiang, P. (2019). *BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer*. Paper presented at the proceedings of the 28th ACM international conference on information and knowledge management (CIKM), Beijing, China.

40. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Polosukhin, I. (2017). *Attention is all you need*. arXiv preprint arXiv:1706.03762.

41. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). *Translating embeddings for modeling multi-relational data*. Paper presented at the neural information processing systems (NIPS), South Lake Tahoe, USA.

42. Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014). *Knowledge graph embedding by translating on hyperplanes*. Paper presented at the proceedings of the 28th AAAI conference on artificial intelligence, Québec, Canada.

43. Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015). *Learning entity and relation embeddings for knowledge graph completion*. Paper presented at the proceedings of the 29th AAAI conference on artificial intelligence, Texas, USA.

44. Ji, G., He, S., Xu, L., Liu, K., & Zhao, J. (2015). *Knowledge graph embedding via dynamic mapping matrix*. Paper presented at the proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing, Beijing, China.

45. Wang, Y., Mohamed, A., Le, D., Liu, C., Xiao, A., Mahadeokar, J., Zhang, F. (2020). *Transformer-based acoustic modeling for hybrid speech recognition*. Paper presented at the proceedings of the 45th IEEE international conference on acoustics, speech and signal processing (ICASSP), Barcelona, Spain.

46. Chen, X., Liu, D., Lei, C., Li, R., Zha, Z.-J., & Xiong, Z. (2019). *BERT4SessRec: Content-based video relevance prediction with bidirectional encoder representations from transformer*. Paper presented at the proceedings of the 27th ACM international conference on multimedia, New York, USA.

47. Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). *BPR: Bayesian personalized ranking from implicit feedback*. arXiv preprint arXiv:1205.2618.

48. Zhang, S., Tay, Y., Yao, L., & Sun, A. (2018). *Next item recommendation with self-attention*. arXiv preprint arXiv:1808.06414.

49. Wang, H., Zhang, F., Xie, X., & Guo, M. (2018). *DKN: Deep knowledge-aware network for news recommendation*. Paper presented at the proceedings of the 2018 world wide web conference.

50. Chetty, R., Friedman, J. N., & Rockoff, J. E. (2014). *Measuring the impacts of teachers II: Teacher value-added and student outcomes in adulthood*. American economic review, 104(9), 2633–2679.

51. Jacob, A. (2012). *Examining the relationship between student achievement and observable teacher characteristics: Implications for school leaders*. International Journal of Educational Leadership Preparation, 7(3), n3.

52. Lu, Y., Zhang, S., Huang, Y., Wang, L., Yu, X., Zhao, Z., & Wu, F. (2021). *Future-aware diverse trends framework for recommendation*. Paper presented at the proceedings of the web conference 2021.

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