Reinforced MOOCs Concept Recommendation in Heterogeneous Information Networks

JIBING GONG, Yanshan University, China
YAO WAN*, Huazhong University of Science and Technology, China
YE LIU, Salesforce Research, USA
XUEWEN LI, Yanshan University, China
YI ZHAO, Yanshan University, China
CHENG WANG, Yanshan University, China
YUTING LIN, Yanshan University, China
XIAOHAN FANG, Yanshan University, China
WENZHENG FENG, Tsinghua University, China
JINGYI ZHANG, Beihang University, China
JIE TANG*, Tsinghua University, China

Massive open online courses (MOOCs), which offer open access and widespread interactive participation through the internet, are quickly becoming the preferred method for online and remote learning. Several MOOC platforms offer the service of course recommendation to users, to improve the learning experience of users. Despite the usefulness of this service, we consider that recommending courses to users directly may neglect their varying degrees of expertise. To mitigate this gap, we examine an interesting problem

* Corresponding authors

Authors' addresses: Jibing Gong, gongjibing@ysu.edu.cn, School of Information Science and Engineering, the Key Lab for Computer Virtual Technology and System Integration, and the Key Laboratory for Software Engineering of Hebei Province, Yanshan University, No. 438 West Hebei Avenue, Qinhuangdao, Hebei, China, 066004; Yao Wan, wanyao@hust.edu.cn, School of Computer Science and Technology, Huazhong University of Science and Technology, Luoyu Road 1037, Wuhan, Hubei, China, 430074; Ye Liu, yeliu@salesforce.com, Salesforce Research, 556 Xixi Road, Chicago, USA; Xuewen Li, lixuewen@stumail.ysu.edu.cn, School of Information Science and Engineering, the Key Lab for Computer Virtual Technology and System Integration, and the Key Laboratory for Software Engineering of Hebei Province, Yanshan University, No. 438 West Hebei Avenue, Qinhuangdao, Hebei, China, 066004; Yi Zhao, zhaoyi.hb@gmail.com, School of Information Science and Engineering, the Key Lab for Computer Virtual Technology and System Integration, and the Key Laboratory for Software Engineering of Hebei Province, Yanshan University, No. 438 West Hebei Avenue, Qinhuangdao, Hebei, China, 066004; Cheng Wang, 862741851@qq.com, School of Information Science and Engineering, the Key Lab for Computer Virtual Technology and System Integration, and the Key Laboratory for Software Engineering of Hebei Province, Yanshan University, No. 438 West Hebei Avenue, Qinhuangdao, Hubei, China, 066004; Yuting Lin, yuting_lin2022@163.com, School of Information Science and Engineering, the Key Lab for Computer Virtual Technology and System Integration, and the Key Laboratory for Software Engineering of Hebei Province, Yanshan University, No. 438 West Hebei Avenue, Qinhuangdao, Hebei, China, 066004; Xiaohan Fang, xiaohan_fang_ysu@163.com, School of Information Science and Engineering, the Key Lab for Computer Virtual Technology and System Integration, and the Key Laboratory for Software Engineering of Hebei Province, Yanshan University, No. 438 West Hebei Avenue, Qinhuangdao, Hebei, China, 066004; Wenzheng Feng, fwz17@mails.tsinghua.edu.cn, Department of Computer Science, Tsinghua University, Beijing, China, 100084; Jingyi Zhang, zhangjy@buaa.edu.cn, School of Cyber Science and Technology, Beihang University, 37 Xueyuan Road, Beijing, China, 100191; Jie Tang, jietang@tsinghua.edu.cn, Department of Computer Science, Tsinghua University, Beijing, China, 100084.

(a) The heterogeneous network in MOOCs   (b) Studying progress   (c) Studying speed
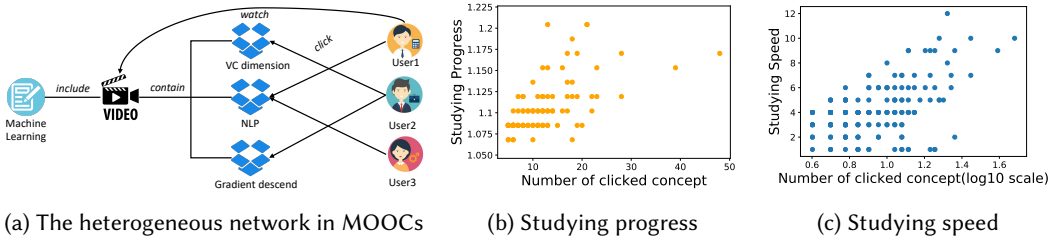
Fig. 1. The motivation of concept recommendation. (a) The heterogeneous network in MOOCs. (b) and (c) denote the progress of studying, and the speed of studying, respectively.

of concept recommendation in this paper, which can be viewed as recommending knowledge to users in a fine-grained way. We put forward a novel approach, termed HinCRec-RL, for Concept Recommendation in MOOCs, which is based on Heterogeneous Information Networks and Reinforcement Learning. In particular, we propose to shape the problem of concept recommendation within a reinforcement learning framework to characterize the dynamic interaction between users and knowledge concepts in MOOCs. Furthermore, we propose to form the interactions among users, courses, videos, and concepts into a heterogeneous information network (HIN) to learn the semantic user representations better. We then employ an attentional graph neural network to represent the users in the HIN, based on meta-paths. Extensive experiments are conducted on a real-world dataset collected from a Chinese MOOC platform, *XuetangX*, to validate the efficacy of our proposed HinCRec-RL. Experimental results and analysis demonstrate that our proposed HinCRec-RL performs well when compared with several state-of-the-art models.

Additional Key Words and Phrases: HIN, reinforcement learning, MOOCs, concept recommendation

1 INTRODUCTION

Massive Open Online Courses (MOOCs), which seek to offer widespread interactive participation and open access over the internet, are quickly becoming a reliable platform for online and remote education. Coursera[1], Udacity[2], and edX[3] are the three most well-known MOOC platforms, which have provided online access to millions of users to study from anywhere in the world. As one of China's leading MOOC platforms, *XuetangX*[4] has more than 1,000 courses available and more than 6,000,000 users worldwide. In MOOCs, *course concepts* are used to describe the knowledge concepts and related subjects covered in the course videos. This can make it easier for users to find the information in the course videos more quickly.

Several approaches of data mining have been developed to discover user preferences and the relationships among users, course topics, and course videos, to enable users to have a better learning experience in MOOCs. These approaches have been used in various related tasks, e.g., behavior prediction [58], course recommendation [31, 88], and user intention understanding [16]. The task most similar to ours is course recommendation, which seeks to recommend relevant courses to users based on their past behavior. From our investigation, we observe that a course is always composed of a series of video lectures, each of which often conveys multiple specific knowledge

---

[1] https://www.coursera.org
[2] https://www.udacity.com
[3] https://www.edx.org
[4] https://www.xuetangx.com

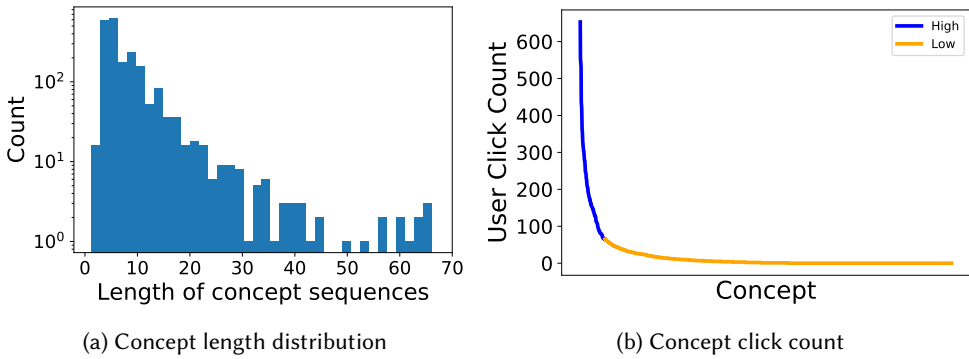(a) Concept length distribution   (b) Concept click count

Fig. 2. A statistical analysis of *XuetangX* dataset.

concepts. Consequently, we argue that recommending courses to users directly ignores the various knowledge levels of users. Figure 1a shows the heterogeneous information network existing in MOOCs. For example, considering the *Machine Learning* course taught by different instructors, the knowledge covered in each course will differ largely (e.g., some instructors will put more focus on the theory of machine learning, whereas others will put more focus on the applications of machine learning in the area of computer science). If a student has a background in statistics theory, he/she may be familiar with several fundamental knowledge concepts, such as *VC dimension* and *gradient descent*, so that he/she will be interested in further applications of machine learning. Additionally, a student who is a beginner in machine learning, he/she may be interested in several fundamental knowledge concepts of machine learning.

Based on this motivation, this paper examines an interesting problem of concept recommendation, with the aim of recommending related topics to users to help them with their online studies. To better illustrate the necessity of this research, we study the effects of clicked course concepts on the progress and speed of studying, as shown in Figure 1b and Figure 1c. The progress of studying is defined as the portion of videos the user has watched in a course, whereas the speed of studying is defined as the number of videos the user has watched in one week. From these two figures, we can see that the number of clicked concepts has a positive correlation with the progress and speed of study. It confirms the fact that clicking relevant knowledge concepts can indeed help the users with their online studies.

For the task of concept recommendation, one primary line of work is based on collaborative filtering. The collaborative-filtering-based approaches provide recommendations according to the historical interactions among users and items. However, this kind of approaches has two major limitations. *(1) Dynamic environment adaptation.* The collaborative-filtering-based approaches consider the procedure of recommendation as a static process, and are intended to maximize the immediate reward of the recommendation, resulting in a fixed greedy recommendation strategy. Therefore, this type of recommendation strategy is sub-optimal and cannot adapt to a dynamic environment where the preferences of users are always dynamic. On the other hand, when representing the historical dynamic preferences of users, it is a challenge to handle the long dependency issue existing in the past click history, as shown in Figure 2a. From Figure 2a, we can observe that the length of concepts clicked by users is mainly between 10 and 30, which is a relatively long sequential dependency when compared with traditional recommendation tasks, such as e-commerce recommendation. *(2) Data sparsity issue.* The interactions among users and concepts are always sparse, particularly as the number of users and concepts is increasingly large. Figure 2b shows the

distribution of user click count for each concept. This distribution follows the long-tail distribution. That is, the majority of concepts are clicked by a few users, which confirms the necessity of concept recommendation. To overcome this issue, several approaches are attempting to include additional supplementary information, such as the social network [78, 96], user/item attributes [65, 79], and contextual information [29, 42], as to enrich the interactions.

These aforementioned restrictions encourage us to develop a model to understand the intents of users, with the ability to recommend personalized concepts in a dynamic interactive environment. We propose in this paper to shape the task of concept recommendation as a sequential recommendation problem. To preserve the dynamic interactions between users and concepts, a reinforcement-learning-based method is proposed by combining exploitation and exploration into a unified framework. Our reinforcement-learning-based recommendation has two advantages: (1) The recommendation agent can adaptively update the trial-and-error strategy during interaction until convergence; (2) The reinforced strategy can produce more diverse recommended results, recommending those items with substantial long-term benefits rather than short-term benefits (i.e., small intermediate rewards). To overcome the data sparsity issue, we observe that there exist heterogeneous relationships among users and concepts, which can be exploited to enrich the side information and boost the performance of recommendation in the following two perspectives: (1) Semantic relationships between knowledge concepts may be provided, which can aid in identifying latent interactions; (2) Users' interest can be understood by tracking the previous records of users alongside these relationships.

This paper puts forward a novel approach (termed HinCRec-RL) for Concept Recommendation in MOOCs, based on Heterogeneous Information Networks (HINs) and Reinforcement Learning. We begin by shaping the concept recommendation as a Markov Decision Process (MDP). We then form the interactions among users, courses, videos, and concepts, into an HIN. We carry out extensive experiments on a real-world dataset obtained from a Chinese MOOC platform, *XuetangX*, to validate the efficacy of the proposed model. Note that this paper is related to our previous work ACKRec [19] for knowledge concept recommendation, which models the entities in MOOCs as an HIN, and proposes an attentional graph convolutional neural network to represent the entities in HIN. To enable the situations of online dynamic interaction, we extend ACKRec by modeling the concept recommendation problem within a reinforcement learning framework.

The primary contributions of this paper include the following three aspects.

- Firstly, we investigate an interesting problem of concept recommendation, which can be considered as a fine-grained recommendation task of course recommendation in MOOCs.
- Secondly, an end-to-end model based on reinforcement learning is proposed to cope with the dynamic interaction among users and concepts, providing a more diverse recommendation. To represent users with sparse data accurately, we propose a meta-path-based user embedding approach with hierarchical graph attention networks.
- Thirdly, we validate the proposed model HinCRec-RL on a dataset collected from a popular Chinese MOOC platform, *XuetangX*. Experimental results and analysis demonstrate the effectiveness of HinCRec-RL when comparing it with several state-of-the-art models.

We structure the rest paper as follows. We first formulate the concept recommendation problem in Sec. 2. We present the details of the proposed approach in Sec. 3. We analyze the *XuetangX* dataset in Sec. 4, and show the experimental results with analyses in Sec. 5. We highlight some related work with a comparison in Sec. 6. We have a discussion of the mechanism of our proposed model in Sec. 7. Finally, Sec. 8 concludes this paper with some potential directions suggested.
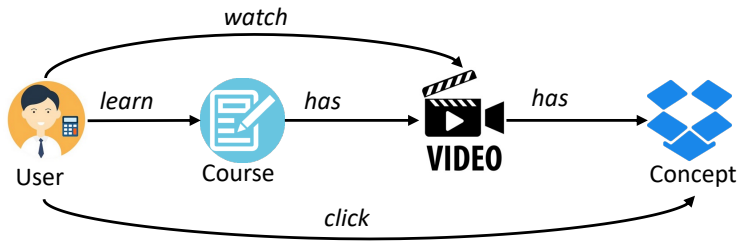
Fig. 3. Network schema of HIN in *XuetangX*.

2 PROBLEM FORMULATION

In MOOCs, suppose that we have U users, C courses, V videos and K concepts. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_U\}$ be the set of users, $C = \{c_1, c_2, \ldots, c_C\}$ be the set of courses, $\mathcal{V} = \{v_1, v_2, \ldots, v_V\}$ be the set of videos and $\mathcal{K} = \{k_1, k_2, \ldots, k_K\}$ be the set of concepts. In reality, a course c_i is always composed of multiple videos, each of which consists of multiple knowledge concepts. From the perspective of users, a user u_j will take a course, watch the videos of the course, and click several related concepts to learn the knowledge presented in the course videos. Consequently, the aim of concept recommendation is to enhance the learning experiences of users by automatically recommending relevant concepts to users based on their preferences acquired from historical experience.

2.1 Recommendation as an MDP

As mentioned before, to adapt to the dynamic environment, we aim to learn a recommendation agent that interacts with users by sequentially recommending knowledge concepts at each time step. We formulate the problem of sequential recommendation as a Markov Decision Process (MDP) consisting of a tuple of five items $(S, \mathcal{A}, P, R, \gamma)$:

- **State space** S denotes the space of user states. Specifically, the state S_t denotes the past preference of user u, observed until time step t. In our scenario, the past preference of user u are the historical learned courses and clicked concepts. More specifically, the state in our scenario represents the user embedding in an HIN.

- **Action space** $\mathcal{A}$ denotes the set of concepts to be selected, i.e., $\mathcal{A} = \mathcal{K}$. Without loss of generality, we sequentially recommend one concept to users at each time step. In each state s_t, an action a_t can be drawn from the available action set $\mathcal{A}(s_t)$, recursively defined as $\mathcal{A}(s_t) = \mathcal{A}(s_{t-1}) \setminus \{a_{t-1}\}$, for $t \neq 0$. This indicates that the agent is not permitted to select those concepts that have been recommended in prior time steps.

- **Reward** R, formulated as $R_{ss'}^a = \mathbb{E}[r_{t+1}|s_t = s, a_t = a, s_{t+1} = s']$, represents the expected immediate reward produced by the environment when state s changes to state s' as a result of action a. In our scenario, the immediate reward of performing an action a is determined by the operation of clicking or not, given by user u. Therefore, we define $R_{ss'}^a = R_{ua}$.

- **Transition function** P, formulated as $P_{ss'}^a = P[s_{t+1} = s'|s_t = s, a_t = a]$, denotes the likelihood that the environment would transit from state s to state s' after performing an action a.

- **Discount factor** γ is used to measure the proportion of reward the reinforced agent considers in the distant future versus those in the immediate future, with a range of $[0, 1]$.
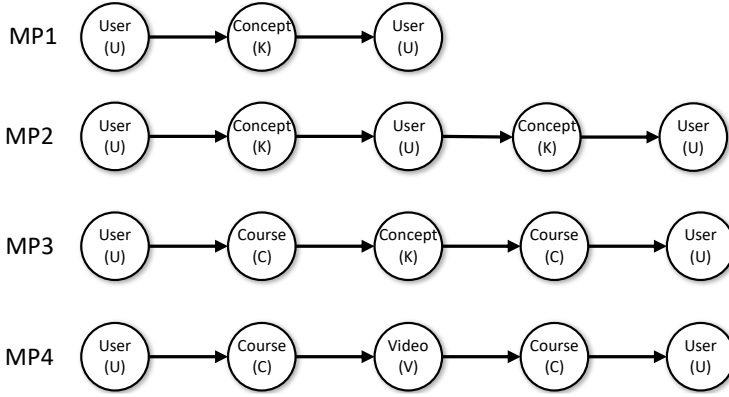
Fig. 4. Defined meta-paths in *XuetangX*.

The concept recommendation task can be formally characterized as follows, using the aforementioned notations and terminologies. Given an MDP $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, the aim of concept recommendation is to find a policy $\pi : \pi \rightarrow \mathcal{A}$ that maximizes the cumulated reward from users, i.e., increasing the clickthrough rate of concepts.

2.2 Construction of Heterogeneous Information Network

In order to model different types of entities (e.g., course, video, concept, and user) and their complex relationships in MOOCs, we resort to considering the MOOC networks as typical heterogeneous information networks (HINs). First, we introduce some preliminary concepts about HINs.

Heterogeneous information network (HIN). Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ denote a graph, which is composed of a collection of nodes $\mathcal{V}$ and edges $\mathcal{E}$. The edges in a graph can describe the relationships/connections among nodes. We refer to the graph as an HIN when the types of nodes and edges are not unique [67]. In our scenario, there are four entities of the HIN in *XuetangX* (i.e., *course* (C), *video* ($\mathcal{V}$), *user* ($\mathcal{U}$), and *knowledge concept* ($\mathcal{K}$)). The relationships among these entities include *learn*, *click*, *watch*, and *has*. Figure 3 describes the schema of network existing in *XuetangX* MOOC. We can extract the meta-path from the network schema, which has been frequently utilized to represent the semantic paths between a pair of entities.

Meta-path. A meta-path is a specific path defined based on the network schema. That is a sequence of relations between two node types, in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$, where A_i denotes the node type [68]. In our scenario, we manually identify four meta-paths as follows to connect users in *XuetangX* MOOC, which will also enrich the semantic representation of each user.

- **MP1 (U_1-K-U_2)** indicates that if two users click the same concept, they are considered to be connected.
- **MP2 (U_1-K_1-U_2-K_2-U_3)** indicates that the two concepts clicked by users 1 and 2, respectively, are both clicked by user 3.
- **MP3 (U_1-C_1-K-C_2-U_2)** denotes that users who take the courses that contain the same concepts will be connected.
- **MP4 (U_1-C_1-V-C_2-U_2)** shows that users are related through paths of courses sharing some common videos.

Table 1. A list of notations used in this paper.

Notation	Description
$\mathcal{G}$	the heterogeneous information network
$\mathcal{V}$	the entity set
$\mathcal{E}$	the relationship set
$\mathcal{SM}$	the network schema
$\mathcal{MP}$	the meta-path set
$\mathcal{P}$	the set of paths sampled based on meta-paths
$\mathcal{S}$	the state of MDP
$R_{ss'}^{a}$	the reward obtained when transferring state s to s' by taking action a
γ	the discount factor
π	the recommendation policy
λ	the weight to regularize the entropy
Φ	meta-path
$\mathcal{N}_i^{\phi}$	meta-path-based neighbors of node i
$\mathbf{h}$	the initial node feature
$\mathbf{u}$	the user embedding
$\mathbf{c}$	the concept embedding

We also show these four meta-paths in Figure 4. In Sec. 5.2, we will investigate the effect of each meta-path and their combinations.

The notations used throughout this paper are listed in Table 1.

3 PROPOSED APPROACH

We describe our proposed methodology in depth in this section, beginning with an overview of the entire framework. To learn the user embedding, we present our graph neural network to represent the introduced HIN. Subsequently, we present our reinforcement learning strategy for recommending concepts to users.

3.1 Overview

Figure 5 depicts the structure of our proposed neural network, which is composed of three sub-modules. (a) Meta-path sampling (Sec. 3.2.1). In this module, we first build an HIN among users, courses, and concepts. Subsequently, we sample meta-paths for each given user using a random walk in the network. (b) Meta-path-based user embedding with a hierarchical attention network (Sec. 3.2.2 and Sec. 3.2.3). In this module, a hierarchical attention network is introduced to embed each math-path into a hidden space. On the node-level, we employ a self-attention layer to represent the user with its neighbors. On the path-level, we apply another attention layer to aggregate semantic representations along each path to the user. (c) Reinforced concept recommendation (Sec. 3.3). In this module, we propose to recommend concepts to users within a reinforcement learning framework. Comparing with other HIN-based representation learning approaches for node classification and recommendation, this is the first time that the HIN and reinforcement learning have been combined to recommend concepts to users in MOOCs.

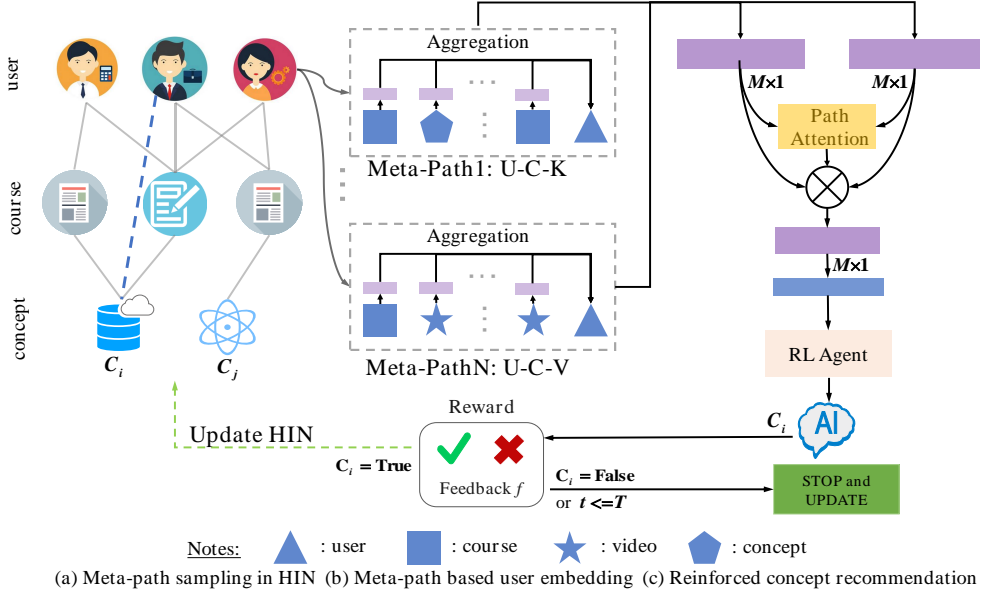(a) Meta-path sampling in HIN (b) Meta-path based user embedding (c) Reinforced concept recommendation

Fig. 5. Overview of the proposed method. (a) Meta-path sampling in the HIN of MOOCs. (b) Meta-path based user embedding with hierarchical attention networks. (c) Reinforced concept recommendation.

3.2 Meta-Path Based User Embedding

We represent the user based on meta-paths. Our intuition is that the representation of users can be enriched by the information aggregated from multi-hop neighbors along the meta-paths. For a given user u_i, we first collect multi-hop neighbors by meta-path sampling, starting from this node. We then design a heterogeneous attention network to aggregate the information along each sampled path.

3.2.1 Meta-Path Sampling. To index the network structure of $\mathcal{G}$, we propose to sample paths on $\mathcal{G}$ based on random walk. Inspired by DeepWalk [57], we start random walk from each user in our training dataset $\mathcal{D}$ for whom a concept will be recommended. Specifically, we first employ a depth-first search, starting from the user node, to sample instances under the restrictive scheme of each meta-path, and set a limited walk length l determining at most how many neighbors would be searched along the sampled path and controlling the size of its neighborhood. Then we utilize a breadth search to repeat the aforementioned procedure N times in parallel for each meta-path, as each path potentially focuses on different input neighbors, which makes it possible to capture rich semantics. Thus, for each user $u_i \in \mathcal{U}$, we sample $|\mathcal{MP}| \times N$ paths. In total, we have $|\mathcal{U}| \times |\mathcal{MP}| \times N$ paths over the HIN. We denote the meta-path as $\mathcal{MP} = \{\Phi_1, \Phi_2, \ldots, \Phi_P\}$ and denote the path corpus as $\mathcal{P} = \{\rho_{\Phi_1}, \rho_{\Phi_2}, \ldots, \rho_{\Phi_P}\}$, where ρ_{Φ_k} denotes a path set sampled based on the meta-path Φ_k.

3.2.2 Node-level Attention. In order to obtain the semantic-specific user embedding, we design a module of node-level attention which first determines the weights of neighbors along a meta-path, and then combines them. As the node types are heterogeneous and various node types have different feature spaces, a simple semantic feature space alignment should be designed. We first project

the node features into a common space. For node i, let $\mathbf{h}_i$ denote its corresponding feature. The projected feature $\mathbf{h'}_i$ can be calculated as follows:

$$\mathbf{h'}_i = \mathbf{M}_{\phi_i} \cdot \mathbf{h}_i, \tag{1}$$

where $\mathbf{M}_{\phi_i}$ is the projection matrix of type ϕ_i. We argue that different nodes for a given meta-path should contribute different weights to the final user embedding. Thus, an attention mechanism is designed to calculate the weight of node i to its meta-path-based neighbors $\mathcal{N}_i^{\Phi}$. Formally, for a given meta-path Φ, the node-level attention weight $\mathbf{a}_{\Phi}$ between nodes i and j can be calculated as follows:

$$\alpha_{ij}^{\Phi} = \frac{\exp\left(\sigma\left(\mathbf{a}_{\Phi}^{\mathsf{T}} \cdot \left[\mathbf{h}'_i \| \mathbf{h}'_j\right]\right)\right)}{\sum_{k \in \mathcal{N}_i^{\Phi}} \exp\left(\sigma\left(\mathbf{a}_{\Phi}^{\mathsf{T}} \cdot \left[\mathbf{h}'_i \| \mathbf{h}'_k\right]\right)\right)}, \tag{2}$$

where $\|$ denotes the concatenation operator, and σ denotes the activation function. Note that, for the sake of simplicity, we randomly select one sampled path of meta-path-based neighbors to calculate the node-level attention. Following [74], we then aggregate the node embedding $\mathbf{u}_i$ for node i along a meta-path Φ. Furthermore, the node-level attention is repeated L times and the attention embeddings are concatenated through a layer of linear projection. Consequently, the user embedding $\mathbf{u}_i^{\Phi}$ along a meta-path Φ is formulated as follows:

$$\mathbf{u}_i^{\Phi} = \|_{l=1}^{L} \sigma\left(\sum_{j \in \mathcal{N}_i^{\Phi}} \alpha_{ij}^{\Phi} \cdot \mathbf{h}'_j\right), \tag{3}$$

where $\|_{l=1}^{L}$ represents the concatenation operation. For a set of meta-paths $\{\Phi_1, \Phi_2, \ldots, \Phi_P\}$, we denote the user embedding along the meta-path set as $\mathbf{u}_i^{\Phi_1}, \mathbf{u}_i^{\Phi_2}, \ldots, \mathbf{u}_i^{\Phi_P}$.

3.2.3 Path-level Attention. Similar to node-level attention, we use attention at the path level. We assign different weights to meta-paths via path-level attention, to obtain the final user representation. The importance of meta-path w_{Φ_k} can be calculated as follows:

$$w_{\Phi_k} = \frac{1}{|\mathcal{P}|} \sum_{p=1}^{|\mathcal{P}|} \mathbf{q}^{\mathsf{T}} \cdot \tanh\left(\mathbf{W} \cdot \mathbf{u}_i^{\Phi_k} + \mathbf{b}\right), \tag{4}$$

where $\mathbf{q}$ represents the path-level attention vector and $|\mathcal{P}|$ denotes the number of meta-path-based sampled paths. That is, we average all the sampled paths for one meta-path to calculate the path-level attention, which expresses sophisticated functions beyond the simple meta-path-based neighbors average. Consequently, the importance of meta-path w_{Φ_k} can be normalized using a softmax layer, i.e., $\beta_{\Phi_k} = \frac{\exp(w_{\Phi_k})}{\sum_{k=1}^{|\mathcal{MP}|} \exp(w_{\Phi_k})}$. The final embedding of user i can be reformulated as the following formula, according to the normalized importance of the meta-path.

$$\mathbf{u}_i = \sum_{k=1}^{|\mathcal{MP}|} \beta_{\Phi_k} \cdot \mathbf{u}_i^{\Phi_k}. \tag{5}$$

The two hierarchical attention modules (i.e., node-level attention and path-level attention) to assign different weights to the node neighbors and different paths have been verified in the node classification task in [74, 75]. We have adopted them in this study with a slight modification. An important difference is that the HIN in our scenario is dynamically updated according to the feedback from the users. That is, a link will be assigned if a user clicks a recommended concept.

3.3 Reinforced Concept Recommendation

As previously mentioned, to simulate the dynamic interactions between users and concepts, we frame the concept recommendation within a reinforcement learning framework. Based on a predetermined loss function, we aim to develop a model to reduce the discrepancy between the model predictions and immediate user responses.

We begin by introducing the simplified setting of concept recommendation within a typical reinforcement learning framework, the goal of which is to find a recommendation policy π_θ that maximizes the expected reward:

$$\mathcal{L}_{rl}(\theta) = \mathbb{E}_{c_t \sim \pi_\theta(\cdot | \mathbf{u}_t)} \sum_{t=1}^{T} r(c_t | \mathbf{u}_t), \tag{6}$$

where $r(c_t | \mathbf{u}_t)$ is the immediate reward. Note that, in contrast to the previous section, the dynamic embedding of user u at time step t is indicated by the subscript t in $\mathbf{u}_t$ in this section. Given a particular user u, if the predicted concept c_t is true, then the reward will be set to 1, otherwise, it will be -1. In particular, $R(c_t | \mathbf{u}_t)$ is the cumulated reward, which can be defined as follows:

$$R(c_t | \mathbf{u}_t) = \gamma^0 R(c_t | \mathbf{u}_t) + \cdots + \gamma^{T-t} R(c_T | \mathbf{u}_T), \tag{7}$$

where γ is the discount factor, which is set in the range from 0 to 1. If the recommended concept c_t is correct, an edge will be established in the current HIN to connect the user and the correct recommended concept. Therefore, we will obtain the new user embedding through the previous HIN embedding step. If the recommended concept is correct, HinCRec-RL will continually recommend a new concept, until it incorrectly recommends a concept or arrives at the max time step T.

However, if the recommended concept is incorrect, no new relation is added, and the structure of HIN $\mathcal{G}$ remains unchanged. Consequently, the user embedding will remain unchanged. As the Q-learning network [70] requires the next step transition s_{t+1} of current step s_t to update, if the recommendation is incorrect, then the predicted Q-value Q_{t+1} will be the same as the target Q_t, and consequently, Q-learning cannot be used in our setting. Therefore, we seek to use the policy gradient to update the model.

We can then learn the optimal policy by using the policy gradient method called REINFORCE [69, 77], a popular learning approach in reinforcement learning. The gradient of the expected cumulative reward can be determined using the following formula, according to the policy gradient method:

$$\nabla_\theta \mathcal{L}_{rl}(\theta) = \sum_{t=1}^{T} \left[\nabla_\theta \log \pi_\theta(c_t | \mathbf{u}_t) R(c_t | \mathbf{u}_t) \right]. \tag{8}$$

In practice, we often observed a collapse onto a suboptimal deterministic policy. To prevent the model from not being able to explore new concepts that could lead to a better recommendation, we add an entropy regularization term to the objective function, as follows:

$$\mathbf{H}[\pi_\theta(c | \mathbf{u})] = \sum_{t=1}^{T} \sum_{c_t \in C} \log \pi_\theta(c_t | \mathbf{u}_t) \pi_\theta(c_t | \mathbf{u}_t). \tag{9}$$

Therefore, the ultimate goal of our model is reformulated as follows:

$$\mathcal{J} = \mathbb{E}_{c \sim \pi_\theta(\cdot | \mathbf{u})} \mathcal{L}_{rl}(\theta) + \lambda \mathbf{H}[\pi_\theta(c | \mathbf{u})]. \tag{10}$$

Here, λ is the regularization weight. The policy gradient directly updates policy π_θ to increase the probability of c_t given state s_t when the reward is positive, and vice versa.

Algorithm 1 depicts the HinCRec-RL algorithm with reinforcement learning. Given the training set $\mathcal{U}_{train}$, clicking data R, the number of time steps T, the number of episodes E, discount factor

Algorithm 1: HinCRec-RL

Input   :Training set $\mathcal{U}_{train}$, clicking data R, number of time steps T, number of episodes E, discount factor γ, and ϵ-greedy parameter ϵ.

Output:The learned recommendation policy π_θ.

1 Meta-path sampling

2 Initialize recommendation policy π_θ with random weights

3 **for** *episode $\leftarrow$ 1 **to** E* **do**

4   Initialize the parameters of HIN $\mathcal{G}$ and user embeddings $\mathcal{U}_{train}$

5   Uniformly pick a user $u_0 \in \mathcal{U}_{train}$

6   Learn the user embedding $\mathbf{u}_1$ by Eq. 5

7 $t \leftarrow 1$

8   **while** *c_t is correct and $t \leq T$* **do**

9   Select the concept c_t using ϵ-greedy policy w.r.t π_θ, observe reward $R(c_t|\mathbf{u}_t)$

10   **if** *c_t is correct* **then**

11   Add a new link between the user u_t and concept c_t on the HIN $\mathcal{G}$

12   Recompute the user embedding $\mathbf{u}_{t+1}$ by Eq. 5

13   **end**

14 $t \leftarrow t + 1$

15   **end**

16   Update π_θ's weights θ according to Eq. 10

17 **end**

γ, and ϵ-greedy parameter ϵ, our goal is to learn the recommendation policy π_θ. We first conduct meta-path sampling and initialize the recommendation policy π_θ, which will be used later. At the reinforcement learning phase, for each episode, we first pick a user and learn the corresponding embedding based on Eq. 5 (see line 4-6). We then dynamically update the policy through reinforcement learning (line 8-15).

3.4 Time Complexity of HinCRec-RL

Here, we analyze the time complexity of HinCRec-RL, as shown in Algorithm 1. The total complexity for a training episode is $O(L \sum_{v \in V} F_v F_1 + LT \sum_{i=1}^{|\mathcal{MP}|}(|E_{\Phi_i}|F_1 + |V_{\Phi_i}|F_1 F_2))$, where L is the number of attention heads, T is the number of time steps of an episode, $|\mathcal{MP}|$ is the number of hand-crafted meta-paths, and F_v is the input dimension of the specific type node. F_1 and F_2 are the output dimensions of node type projection and path-specific transformation, respectively. $|E_\Phi|$ and $|V_\Phi|$ are the numbers of sampled-path-based node pairs and user nodes, respectively. More specifically, the time complexity of a single node-level attention head can be expressed as $O(|V| + |E|) = O(\sum_{v \in V} F_v F_1 + |E_\Phi|F_1)$, including input feature transformation, the computation of sampled path attention coefficients, and node-level aggregation. The time complexity of the path-level attention can be expressed as $O(|V|) = O(|V_\Phi|F_1 F_2)$, including path-specific transformation and meta-path based sampled path aggregation. Note that we only need to pick one user for each episode uniformly and recompute its user embedding iteratively during sequential recommendation training with reinforcement learning. Within an episode, the action space of ϵ-greedy influences the number of time steps, whereas the requirement to converge is correlated to the action space. Thus, HinCRec-RL is highly computationally efficient, and the total complexity is linearly proportional to the number of nodes and meta-path-based node pairs.

Table 2. The description of *XuetangX* datasets.

Nodes	Count	Links	Count
concept	2,527	concept-course	21,507
		concept-video	11,732
user	3,111,637	user-course	15,045,219
		user-video	53,481,869
course	7,327	course-concept	69,012
		course-video	811,841
		course-user	16,724,852
video	62,191	video-course	247,433
		video-concept	11,732
		video-user	53,971,707
Total	**3,183,682**	**Total**	**140,096,904**

4 EXPERIMENTAL SETUP

We carry out experiments by addressing the following research issues to assess the performance of our proposed model HinCRec-RL.

- **RQ1:** How effective is the proposed model HinCRec-RL for concept recommendation when compared with baseline methods?
- **RQ2:** How do various involved meta-paths and their combinations affect user embedding?
- **RQ3:** How dose the proposed model HinCRec-RL perform when the hyper-parameters are changed?

4.1 Experimental Dataset

To conduct our experiments, we construct training and test datasets using the data from *XuetangX* MOOC platform, where the training set spans from 1st October, 2016 to 30th December, 2017, whereas the test set spans from 1st January, 2018 to 31st March, 2018. We pair a target course with all historical enrolled courses to form a sequence which is further considered as an instance of our datasets. For each instance when conducting the training phase, we consider the final course in the sequence to be the target course and the remaining to be the historical courses. When conducting the test process, for a user, a target concept in the test set indicates a concept that must be involved in the enrolled courses, and historical concepts indicate the corresponding concepts to the same user in the training set. On the other hand, the same concept label is usually clicked by a user, and thus, multiple clicked concept records are generated. In our experiments, we serialize these records into a single one. We then paired each positive instance with a random sample of 99 negative ones in the test set.

Overall, the obtained dataset contains 62,191 videos, 7,327 courses, 3,111,637 users, 2,527 concepts and 140,096,904 relations between these entities. The description of this dataset is presented in Table 2.

4.2 Evaluation Metrics

We pick several metrics commonly used in recommender systems to evaluate the proposed model. They are Mean Reciprocal Rank (*MRR*), Normalized Discounted Cumulative Gain of top-K items (*NDCG@K*) and Hit Ratio of top-K items (*HR@K*). We choose 5, 10 and 20 for K in our experiments.

Table 3. Quantitative results (%) of experiments on *XuetangX* MOOCs datasets. (Best scores are in boldface.)

Model	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20	MRR	AUC
BPR	41.50	56.28	72.12	28.95	33.74	37.76	28.68	84.34
MLP	32.75	52.66	69.88	17.67	25.32	30.71	18.34	82.73
FM	36.03	54.40	72.39	23.99	29.89	34.44	24.46	83.20
FISM	36.41	54.31	70.79	20.15	27.73	33.83	21.65	84.24
NAIS	36.47	57.92	77.99	21.93	28.10	34.12	22.15	86.76
NASR	38.67	53.21	67.13	21.19	27.84	32.93	19.69	79.90
HinCRec-SL	60.73	78.41	**90.04**	40.89	46.62	49.60	37.85	**92.10**
HinCRec-RL	**66.21**	**81.53**	88.74	**46.21**	**51.35**	**53.22**	**42.82**	89.64

HR@K is used to measure the proportion of ground-truth instances which are correctly recommended among the top-*K* items. This is a recall-based metric and can be formulated as

$$HR@K = \frac{\#Hits@K}{|GT|},\tag{11}$$

where $|GT|$ is the size of the test dataset. *HR@K* measures the percentage of successes in prediction to the number of attempts. *NDCG* is usually applied to assess a retrieval syetem's performance when considering the graded relevance of retrieved entities. Thus, it is also used as another ranking metric in our experiments and can be defined as

$$NDCG@K = \frac{1}{|Q|} \sum_{q=1}^{|Q|} Z_{kq} \sum_{j=1}^{k} \frac{2^{r(j)} - 1}{\log(1 + j)},\tag{12}$$

where $r(j)$ denotes a relevance score calculated for a retrieved entity for query q. Z_{kq} indicates a normalization factor and is usually used to ensure that the *NDCG* for a perfect ranking at k for query q is 1. *MRR* is used to measure the ranking of our searched results and can be defined as

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i},\tag{13}$$

where $|Q|$ indicates the size of a query set.

Moreover, we leverage Area Under the ROC curve (*AUC*) to measure the quality of recommendation ranking.

4.3 Implementation Details

The proposed model is trained on an Nvidia GeForce GTX1080Ti GPU with 11 GB of RAM. The dataset is split into two portions for our evaluation tasks. The first portion contains 80% instances for training and the second portion contains 20% instances for testing. We choose 64 as the dimension size of the final user embedding. First, a cross-entropy loss function with 10,000 episodes is leveraged to pre-train the model. The learning rate is set to 0.001, and the mini-batch size is set to 8. Subsequently, the learning rate is reduced to 0.0001 and the loss function is adjusted to the policy gradient. Furthermore, the weight of the regularization term λ is set to 0.08 to prevent the model from overfitting. The sequence length of recommended concepts is automatically maintained consistent with the mini-batch size. Through the entire training of the HinCRec-RL model, we update the parameter using the Adam optimizer [37].

Table 4. Different results (%) from different combinations of meta-paths. (Best scores are in boldface.)

Meta-Path	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20	MRR	AUC
MP1	63.51	75.23	**90.99**	43.08	46.89	50.90	39.24	91.89
MP2	45.33	61.68	79.44	29.59	34.90	39.43	28.44	86.92
MP3	46.15	67.42	84.62	30.37	37.27	41.66	29.76	88.69
MP4	50.68	69.86	86.76	34.32	40.53	44.78	33.01	91.32
MP1 & MP2	63.47	77.17	89.04	44.33	48.83	51.81	41.04	**93.15**
MP1 & MP3	60.55	76.15	87.61	43.37	48.42	51.39	40.96	93.12
MP1 & MP4	65.02	78.48	88.34	45.69	50.02	52.48	42.15	89.69
MP2 & MP3	46.98	68.37	81.86	29.91	36.78	40.23	28.64	89.30
MP2 & MP4	53.02	73.02	89.30	35.68	42.09	46.20	33.94	92.09
MP3 & MP4	47.44	65.58	84.65	32.30	38.16	42.97	31.56	86.51
MP1 & MP2 & MP3	57.73	75.45	89.55	41.06	46.86	50.47	39.26	91.82
MP1 & MP2 & MP4	59.35	77.23	89.76	43.24	49.12	51.38	41.36	89.12
MP1 & MP3 & MP4	61.26	80.82	88.58	44.73	50.90	52.84	42.37	89.04
MP2 & MP3 & MP4	38.01	60.18	80.09	23.87	30.93	35.85	24.00	90.04
MP1 & MP2 & MP3 & MP4	**66.21**	**81.53**	88.74	**46.21**	**51.35**	**53.22**	**42.82**	89.64

5 EXPERIMENTAL RESULTS

In this section, we respond to the aforementioned research issues through conducting experiments and analyzing the results.

5.1 Comparison with Baselines (RQ1)

We compare the proposed model with the following baselines in order to evaluate its performance.

- **BPR** [61] obeys a Bayesian method to minimize the pairwise ranking loss for recommendation tasks.
- **MLP** [22] utilizes a multi-layer perceptron to compute the recommendation probability by leveraging embedding pairs of user and concept.
- **FM** [60] is a typical matrix decomposition method that can easily address the problem of feature combination. For fairness, we use only the embeddings of users and concepts in our experiments.
- **FISM** [32] is a type of CF algorithm. It can utilize the average historical concept embedding and the target one to perform recommendation tasks.
- **NAIS** [21] is another type of CF algorithm. It can differentiate among the importance of all historical concepts by utilizing the attention mechanism.
- **NASR** [39] is considered an improved GRU model, and can calculate an attention coefficient for each historical concept by using the corresponding hidden vectors produced by GRU.
- **HinCRec-SL** represents the proposed model that only applies an HIN for user embedding, without reinforcement learning.
- **HinCRec-RL** represents the proposed model that combines the HIN for user embedding and the reinforcement learning for better recommendation.

We conduct comparative experiments between our proposed HinCRec-RL model and several baseline ones using offline datasets. According to the comparative experiments results reported in Table 3, HinCRec-RL largely outperforms all other baselines in terms of *HR@K*, *NDCG@K*, *MRR* and *AUC*. We also compare the HinCRec-SL/RL model with these baselines and observe that the proposed HIN contains more useful semantic information of users. When comparing HinCRec-SL with HinCRec-RL, we can observe that the proposed reinforcement learning also has a positive effect on the performance based on supervised learning. For example, in terms of *HR@5* and *NDCG@5*, HinCRec-RL is more than 10% better than HinCRec-SL. This can be illustrated by the
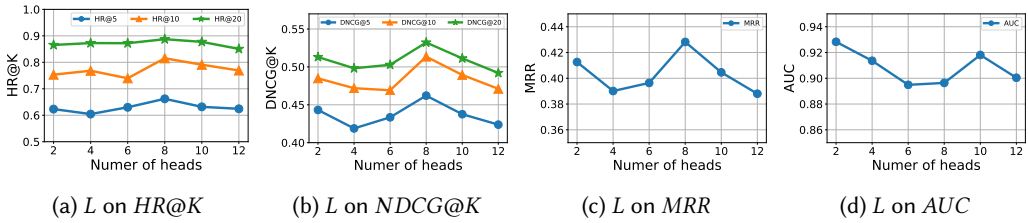
Fig. 6. Parameter sensitivity of HinCRec-RL over attention head L in HIN embedding.

fact that the exploration mechanism in reinforcement learning can yield a better optimized result with our model. More specifically, HinCRecRL achieves an $HR@5$ of 66.21%, an $HR@10$ of 81.53%, an $HR@20$ of 88.74%, an $NDCG@5$ of 46.21%, an $NDCG@10$ of 51.35%, an $NDCG@20$ of 53.22, and an MRR of 42.82% on the the dataset of *XuetangX* MOOCs. Although HinCRec-RL performs slightly worse than HinCRec-SL in terms of $HR@20$, it shows significant improvements when compared with the baselines (i.e., 9.74%-27.54% increment in terms of $HR@5$, $NDCG@5$, MRR and AUC). We attribute such performance enhancements to the fantastic HinCRec-RL modeling: (1) The technique based on the hierarchical graph attention aggregation guided by the user's meta-paths is capable of encoding the interrelationship between concepts and users, hence providing a richer and more comprehensive representation learning of concepts and users. (2) Benefiting from the representation learning, HinCRec-RL can preserve the user's state information and collect more informative rewards from the reinforcement learning process. (3) HinCRec-RL first combines the HIN and reinforcement learning to recommend knowledge concepts for users' course learning in MOOC platforms. In contrast, all the baselines ignore this strategy.
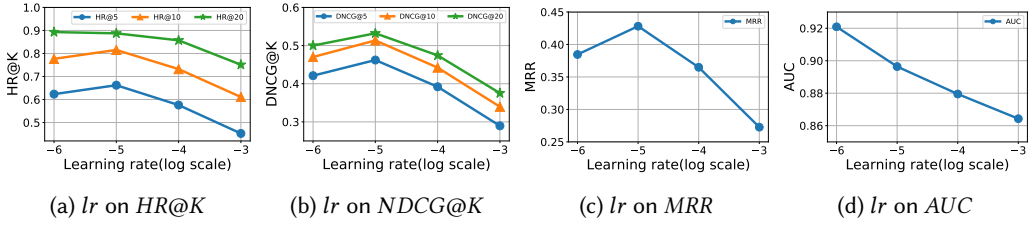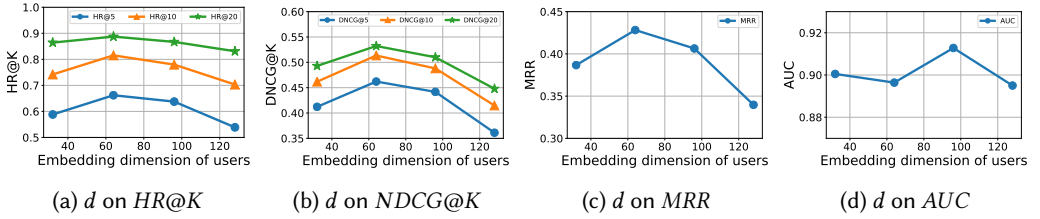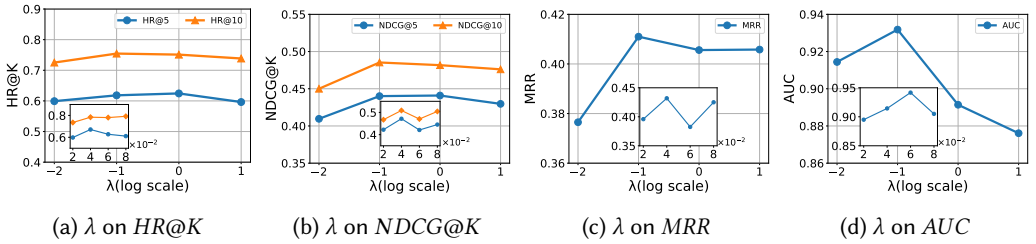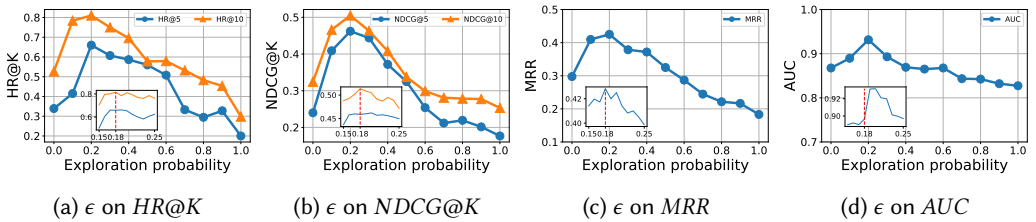
5.2 Effect of Different Meta-Paths (RQ2)

We examine various meta-paths and their combinations to demonstrate the effectiveness of these heterogeneous graph patterns in improving the performance of HinCRec-RL. More specifically, to describe the relatedness among pairs of users, four kinds of meta-paths are chosen for our study. These meta-paths are MP_1: U_1-K_1-U_1, MP_2: U_1-K-U_2-K-U_3, MP_3: U_1-C_1-K-C_2-U_2, and MP_4: U_1-C_1-V-C_2-U_2. The combination of some single or simple meta-paths (e.g., MP_1) may affect our model's performance. To illustrate this, we conduct related experiments and show the results in Table 4. Each single meta-path (i.e., MP_1, MP_2, MP_3, MP_4) and the combinations thereof contribute different performances to the proposed model. Thus, we can conclude that different single meta-paths can result in nonnegligible effects on HinCRec-RL's performance. It is apparent that the adjacent combination of single or simple meta-paths has a positive correlation property. Among the four meta-paths, $MP1$ can achieve a high score in several metrics; thus, it can be considered the best designed meta-path for user representation. This is appropriate since the interests of users are usually affected by their neighbors. Moreover, the more meta-paths the combinations include, the better is the model performance. The most convincing evidence to illustrate this aspect is that the combination of all the four meta-paths causes the best performance of the proposed model.

5.3 Parameter Analysis (RQ3)

We study parameter sensitivity and present the results of HinCRec-RL for diverse parameters in Figures 6-9.

5.3.1 Impact of Attention Head L in HIN Embedding. We analyze the performance of HinCRec-RL with varying numbers of attention heads, proving the validity of multi-head attention involved

Fig. 7. Parameter sensitivity of HinCRec-RL over learning rate *lr*.

(a) *lr* on *HR@K*   (b) *lr* on *NDCG@K*   (c) *lr* on *MRR*   (d) *lr* on *AUC*

(a) *d* on *HR@K*   (b) *d* on *NDCG@K*   (c) *d* on *MRR*   (d) *d* on *AUC*

Fig. 8. Parameter sensitivity of HinCRec-RL over embedding dimension *d* in HIN embedding.

(a) λ on *HR@K*   (b) λ on *NDCG@K*   (c) λ on *MRR*   (d) λ on *AUC*

Fig. 9. Parameter sensitivity of HinCRec-RL over λ.

(a) ϵ on *HR@K*   (b) ϵ on *NDCG@K*   (c) ϵ on *MRR*   (d) ϵ on *AUC*

Fig. 10. Parameter sensitivity of HinCRec-RL over the probability of exploration ϵ.

in HIN embeddings. In sequence, there are 2, 4, 6, 8, 10 and 12 attention heads in our analytical design. The experimental results are given in Figure 6. We can see that the increase of the number of heads slightly elevates the model's performance. Although the attention mechanism included in our model requires slightly longer training time, the results prove that it can make our model more reliable and robust.

5.3.2   Impact of Learning Rate lr. We tune *lr* to learn a recommendation policy better. The impact of *lr* on HinCRec-RL is described in Figure 7. We can see that, when increasing the learning rate
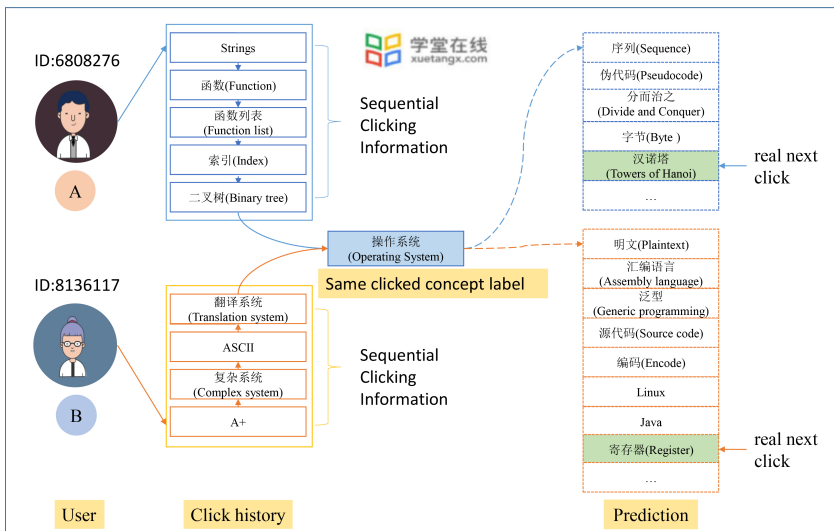
(a) Case 1
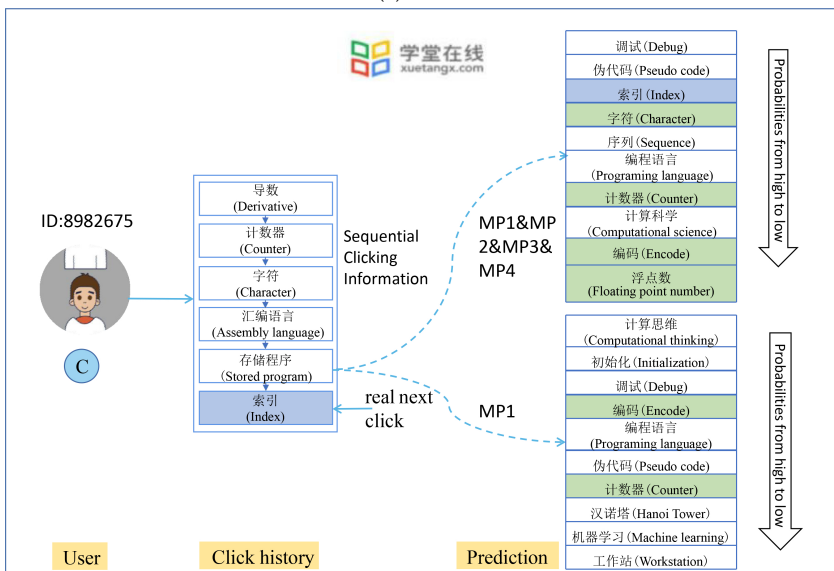
(b) Case 2

Fig. 11. A case of the diversity of concept recommendation, which contains a sequence of clicked concepts within *XuetangX*'s dynamic learning environment.

from 10*e*-6 to 10*e*-5, the performance is improved in terms of the *HR@K*, *NDCG@K*, and *MRR*, whereas the performance is degraded when increasing *lr* further. Therefore, we can conclude that the learning rate affects HinCRec-RL. The best *lr* is approximately 10*e*-5.

5.3.3   Impact of Embedding Dimension d. Aiming to investigate the embedding dimension settings of entities, we conduct experiments with various numbers of dimensions (i.e., 32, 64, 96 and 128). The experiment results are shown in Figure 8. When increasing *d* from 32 to 64, the performance is

improved in terms of $HR@K$, $NDCG@K$, and MRR, whereas the performance is degraded when increasing d further. Our final settings have the user and knowledge concept vector dimension set to 64. In light of the experiment findings, we can conclude that a reasonable embedding dimension is crucial to enhance recommendation performance in our task.

5.3.4   Impact of Regularization Rate λ. We conduct experiments on entropy regularization to investigate the efficacy of the parameter λ. We set λ to 0.01, 0.1, 1 and 10, and show the corresponding experimental results in Figure 9. In addition, for more fine-grained analysis, we set λ to 0.02, 0.04, 0.06 and 0.08, and present the results in a zoomed view. Figure 9 shows that, when increasing λ from 10e-2 to 10e-1, the performance is improved in terms of the $HR@K$, $NDCG@K$, MRR and AUC, whereas the performance is degraded when increasing λ further. Therefore, we can conclude that λ affects the performance of HinCRec-RL, and the best λ is approximately 10e-1.

5.3.5   Impact of Exploration Probability ϵ. We investigate the random action probability in an ϵ-greedy policy. We set the range of ϵ range from 0 to 1 in steps of 0.1, and present the corresponding results in Figure 10. When increasing ϵ from 0 to 0.2, the performance is improved in terms of the $HR@K$, $NDCG@K$, MRR and AUC, whereas the performance is degraded when increasing ϵ from 0.2 to 1 further. The thumbnail of Figure 10 shows that HinCRec-RL performs optimally when ϵ is approximately 0.18. Therefore, the exploration probability affects the performance of HinCRec-RL.

5.4   Case Study

Figure 11 shows a case study to examine the availability of HinCRec-RL, particularly its ability to adapt to dynamic environments when providing personalized concept recommendations. User A (ID: 6806276) and User B (ID: 8136117) are randomly chosen from the testing dataset. They have different educational backgrounds and consequently experience different learning paths in the *XuetangX* MOOC platform. Figure 11a first shows the clicked history of the two users and then displays the concept lists recommended to them separately. From this figure, we can observe that User A is learning a course named "*Introduction to Algorithms*", whereas User B is learning a course named "*Natural Language Processing*". Both users selected distinct knowledge concepts pertaining to the same target course "*Operating System*". In this case, our model can still adaptively consider the currently selected concepts, and creatively recommend the next one depending on the past. We attribute it to the reinforcement learning mechanism we have adopted, i.e., the proposed approach can dynamically interact with the environment.

On the other hand, we also show a case of recommending different concept lists for a given user according to different meta-paths, as shown in Figure 11b. For User C with ID 8982675, given his click history, we show the recommended results by meta-paths $MP1\&MP2\&MP3\&MP4$ and $MP1$, respectively. The result recommended by $MP1\&MP2\&MP3\&MP4$ is more precise than that recommended by $MP1$. We attribute this to the fact that the meta-paths can aggregate the information related to users from neighbors, as data augmentation. This also demonstrates that introducing heterogeneous graph information into user embedding is effective.

6   RELATED WORK

We will present the relevant studies from three aspects, namely, mining in MOOCs, HIN learning, and recommender systems.

6.1   Mining in MOOCs

MOOCs can be viewed as interactive systems containing large-scale multimedia objects, user interaction records, and HINs. Several data mining techniques have been employed to mine these plentiful resources, to improve the user experience on online learning platforms. These data can

also be exploited to reveal, in detail, how students behave and how learning takes place. Existing studies on MOOCs fall under two major categories. One line of work focused on the course content, such as course concept extraction [45, 50], course concept expansion [84], and prerequisite relation discovery [49, 83]. It is common to learn how the representations of entities leverage both context information and content. In addition to the title and description of each course, other types of entities, such as teachers, videos, and concepts, can be constructed into heterogeneous networks to explore more corresponding beneficial semantic relationships. Pan et al. [49] explored the prerequisite relations among concepts by leveraging the concept semantics of course , the video context of course and the structure of course. The other line of work pays attention to the learning behavior patterns of students [58]. For example, some studies [31, 88] recommended courses to students. Additionally, some studies [16, 40] predicted dropouts, whereas other studies [33] predicted the performances of students. Diverse students' behaviors are essential for MOOC research because they are meaningful to mine the motivations [16], cognitive abilities [8], and social activities [34] of students static course research and contents. Yu et al. [85] collected student behaviors data from MOOCs, which indicates the relationships between courses and concepts. These records include student profiles, video watching behaviors, exercise behaviors, comments and replies.

Among the related studies, course recommendation is the task most relevant to ours. Our study not only deploys concept recommendation in the dynamic interactive environment, but also puts more focus on modeling the dynamic interaction among users and concepts, and enriching the representation of users based on heterogeneous information networks. Another related work to ours is [19], which proposed an attentional graph convolutional network for concept recommendation. Different to [19], we resort to modeling the dynamic interactions among users and concepts by reinforcement learning.

6.2 Learning with Heterogeneous Information Networks

Many applications, e.g recommender systems [24], social event detection systems [4, 53], etc, capture the meaningful semantic and structural information by constructing an HIN in auxiliary data. Specially, lots of entity types (e.g., users, courses, and knowledge concepts) and rich edges among them which naturally constitute an HIN, can be easily found in E-commerce applications or online platforms (e.g., Amazon, eBay and *XuetangX* MOOC).

HIN-based methods have outperformed the other previous methods in mining more meaningful information owing to their excellent ability. Sun et al. [68] suggested using meta-paths to explore the information network's meta-structure which can systematically explore fruitful semantic relationships across several kinds of objects. Metapath2vec [13] generated the meta-path by utilizing skip-gram and random walk to represent the HIN. Cai et al. [3] proposed GAN-HBNR which is based on generative adversarial bibliographic network module for citation recommendation. Cen et al. [5] introduced an embedding method for the attributed multiplex heterogeneous network to boost the behavior of the recommender system. Chen et al. [7] proposed a novel metric learning based heterogeneous information network embedding model named PME which uses a unified technique to capture 1st-order and 2nd-order proximities. HAN [74] introduced the attention mechanism into an HIN and achieved a comparable performance in node classification and clustering. Sun et al. [66] presented a heterogeneous hypergraph learning model based on graph neural networks to characterize multiple non-pairwise relations. Furthermore, the attention mechanism has been adopted in several approaches, such as HERec [63] and MCRec [25].

MOOCs are more typical for utilizing the emerging HINs because they contain richer links and semantics in objects, and are thus a fresh direction for data mining. We construct a framework by combining extended deep reinforcement learning with the representations of user status to provide recommendations of relevant knowledge concepts in this study.

6.3 Recommender System

Recent recommendation works can be grouped approximately into four categories [1, 43, 52]: collaborative-filtering (CF) based recommendation, content-based (CB) recommendation, conventional hybrid recommendation, and deep-learning-based recommendation.

6.3.1 CF-based Recommendation. CF has become one of the most efficient recommendation algorithms because of its simplicity and efficiency. It aims to recommend items on the basis of historical data of user–item interactions [1]. Recommender systems rely on either explicit or implicit feedback, which contains abundant information about the interests of users [86]. There are two kinds of algorithms deriving from CF, namely user-based CF (UserCF) and item-based CF (ItemCF) [18]. The major challenges of CF approaches are data sparsity issues [82] and the cold-start problem [97]. In order to alleviate these problems in MOOC course recommendation, Pang et al. [51] proposed MLBR, which uses learner-course vectors rather than a huge and sparse learner-course matrix. Rabahallah et al. [59] combined ontology and CF to recommend personalized courses to online students, where ontology can provide a semantic description of students to mitigate the cold-start problem.

6.3.2 Content-based Recommendation. Compared with CF, CB prioritizes the ratings of the target users and the attributes of items liked by them in the past. Currently, several CB approaches have been proposed for course recommendations on MOOCs. Huang and Lu [27] utilized feature modeling for items and user profiles constructing and matching the item features with the user's preferences. In comparison to CF, the CB suggestion does not require a substantial rating matrix and offers the benefits of user independence and the capacity to suggest new goods. Although CB approaches are significantly better at proposing new things, they cannot yet deliver services to new system users.

6.3.3 Hybrid Recommendation. Hybrid models seek to get the best of both worlds by combining both the basic and the enhanced methods [1]. They can incorporate the advantages of different types of recommendation methods to create a unified method that is more robust in a wide range of contexts. Typically, basic models contain CF and CB methods, whereas enhanced ones include factorization machine (FM) [44, 46], GBDT [35], and other combined variants. Classical methods include GBDT+LR [23] and LS-PLM [17]. The GBDT+LR model has the ability to perform higher-order feature combination. It promotes the trend of feature engineering modeling. The LS-PLM model is a three-layer neural network with an attention mechanism, which has a stronger expression. Several related studies leverage hybrid models for online course recommendation [9, 36]. Wan and Niu [73] proposed an approach called SI-IFL for recommending courses to learners, which integrates sequential pattern mining, the learner influence model, and self-organization-based recommendation strategy. Although deep learning has demonstrated considerable potential for learning effective representations, traditional recommendation models still retain irreplaceable advantages owing to their strong interpretability and fast training speed. In general, there are complicated relationships between traditional recommendation models and deep learning models.

6.3.4 Deep-learning-based Recommendation. Compared with traditional recommendation methods, deep learning recommendation methods have deeper neural network layers, and hence, they can explore more complex connections between users and items.

Graph neural networks (GNNs), particularly dynamic GNNs [54], graph convolutional networks (GCNs) [76], graph inductive representation learning (GraphSAGE) [20], graph attention networks [71], and graph isomorphism networks [81], have gained considerable attention. The main idea behind GNN is to aggregate the neighbor features of nodes by introducing convolution.

For example, MixGCF [28] introduces negative samples by using both positive mixing and hop mixing methods, which leverages the information of several local graphs to train GNN-based CF recommendation models. SHCF [38] is a novel sequence-aware graph-based CF method for recommendation that combines both high-order collaborative messages and sequential patterns. Deng et al. [12] predicted paper review rating via a hierarchical network (HabNet) that leverages a bidirectional self-attention network. Fan et al. [14] proposed a heterogeneous graph network through a meta-path-guided method to model complex entities and rich edges for intent recommendation. Jin et al. [30] proposed an interaction model considering the information of neighborhoods for recommendation in heterogeneous views. Another study [89] presented a comprehensive analysis on network embedding methods for recommendation. Further, the interpretability [6] and privacy of recommender systems [2, 47] have been investigated.

Recently, reinforcement learning has been widely utilized in several fields; for example, OpenAI's AlphaStar [72] defeated professional players in StarCraft II. Silver et al. [64] proposed a reinforcement learning method to build a system for customer interaction. Peng et al. [56] employed a multi-agent RL algorithm to select optimal aggregation strategies in a multi-relational GNN for social event detection tasks. Especially in recommender systems, as a method that focuses on how the agents act in the environment, reinforcement learning enables the recommender to capture the real-time feedback of users effectively for learning optimal recommendation strategies. In practice, several studies have begun to regard recommendation as a deep reinforcement learning problem, and have achieved comparable performances in movie recommendation [41, 92], news recommendation [62, 95], music recommendation [48], e-commerce [93], and health care [90].

More specifically, Zhou et al. [96] proposed a knowledge-graph-enhanced RL method on an interactive recommender system to deal with the sample efficiency issues. Chen et al. [10] introduced the attention method to aggregate the state vector of knowledge graphs and then calculated the Q value in the user–item interest graph through graph convolution. Zou et al. [98] introduced a novel Q network to explore user's long-term interest. On this basis, they also proposed a RL framework [99] to guarantee stable convergence and low time complexity. Huang et al. [26] proposed a method using RNN and RL to improve the accuracy of long-term recommendations. Yu et al. [87] designed VL-Rec, a vision-language recommendation framework with an attribute-augmented RL, which provides natural language feedback to perform effective interactions with the user. Zhao et al. [91] proposed MaHRL based on hierarchical RL for recommendation. The low-level agents learn the short-term preferences of users while the high-level agents focus on learning the uses' long-term preferences and guide the low-level agents. Xie et al. [80] addressed the integrated recommendation task which also employed a hierarchical model. Zhang et al. [88] removed noisy courses from historical course sequences and then performed hierarchical reinforcement learning. Feng et al. [15] and Zhao et al. [94] combined two recommendation scenarios which contain the homepage and product detail page into one recommendation target on an e-commerce platform. Peng et al. [55] proposed to address the complexity of networks by designing a novel neighborhood-selection-guided multi-relational GNN, which maintains relation-dependent representations simultaneously.

Different from existing recommendation methods based on reinforcement learning [11, 48], our model leverages rich semantic meta-path-based context, which can learn better representations by specific interactions for students, courses, concepts, items, etc.

7 DISCUSSION

In this section, we discuss our study from three aspects: comparison with our previous model, verifying the effectiveness our model, and presenting the threats to validity.

7.1 Comparison with Our Previous Work

We revisit our previous model ACKRec [19] for knowledge concept recommendation, and compare it with HinCRec-RL.

Attention-based GCNs versus Hierarchical Attention Networks. By combining information from neighbors based on meta-paths, our models can learn the node representation in heterogeneous graphs. In node-level aggregation, we use information propagation in GCNs within meta-path-based neighbors in ACKRec, which sets the symmetric normalization term between two nodes as $1/\sqrt{|\mathcal{N}_i||\mathcal{N}_j|}$. This encodes the prior that the intermediate representation of node $j \in \mathcal{N}_i^{\Phi_k}$ to node i under a single meta-path Φ_k exploits the proximity structure of the graph. HinCRec-RL, in contrast to GCN, uses self-attention to learn the importance of $\alpha_{ij}^{\Phi_k}$ among meta-path-based neighbors. By assigning weights to the meta-path-based neighbors, HinCRec-RL may be applied to graph nodes with varied degrees. Both of our models use the attention mechanism during path-level aggregation to combine the representations of entities learnt while following various meta-paths and produce the final attentional joint representation. Experimental results show that, in the representation learning stage, both ACKRec and HinCRec-RL can both fully utilize the structure of HINs to learn excellent node embeddings. This is useful for the further downstream recommendation task.

Extended Matrix Factorization versus REINFORCE. We feed both content and context information into the basic matrix factorization model to form our previous, robust and explainable model, ACKRec. It adopts gradient descent to obtain a scalar product of user latent vectors $p_k \in \mathbb{R}^d$ and knowledge concept ones $q_u \in \mathbb{R}^d$ plus users representations e_u and knowledge concepts e_k approximate original ratings r_{uk}. Although ACKRec only slightly alleviates the classic data sparsity problem, it is no longer useful in incremental recommendation settings. The fundamental problem with such a system is that it continuously welcomes new users and concepts. The recommender system for top-K concepts used at *XuetangX* is implemented in practice by the HinCRec-RL model presented in this paper. When the system is running, it may concurrently learn the interests of the users and utilize those interests. This is distinct from the offline setting of ACKRec. In this study, we apply REINFORCE to enable HinCRec-RL to adapt to the dynamic interaction environment between users and concepts, where simultaneous exploration and exploitation of the search space takes place. In the simplest case, the recommendation agent may have to decide the concept lists to recommend. The agent would either earn a reward for the success of the recommendation or receive a penalty when a user clicks the particular notion on the list. Thus, the online recommendation policy network changes can be performed in real time. Experimental results show that HinCRec-RL can perform well in dynamic environment. We will run HinCRec-RL stably in a live production system in the future.

7.2 Why does HinCRec-RL Work?

Here we summarize three main features of HinCRec-RL that may explain its effectiveness in concept recommendation in MOOCs.

Mining Latent Information in Heterogeneous Graphs. Traditional approaches aimed at solving the data sparsity issue by supplementing information about the context and its own attributes are not work effective in our scenario owing to sparse interactions between users and knowledge concepts coupled with a lack of attributes. To enhance the relatedness between users and concepts, we introduce course and video entities and build a heterogeneous graph as a bridge for message passing from concepts to users. In addition to direct interactions between users and concepts, the heterogeneous graph contains rich latent semantic information which is not easy to mine but is crucial.

Attentional Multiple Meta-path-based Node Aggregation Representations. HinCRec-RL learns user embedding representation by integrating the representations of other adjacent nodes, which are those nodes on a path selected based on a specific meta-path schema in the heterogeneous graph. In the model, node weights and meta-paths are included in addition to node properties. Therefore, it can capture the fine-grained connection between users and concepts better and interpret users' preferences by tracking and tracing the users' historical behaviors with the node relationships in these meta-paths.

Dynamic Interaction through Reinforcement Learning. In several real-world scenarios, a user's interest always changes rapidly with time, which requires methods that can react quickly to such changes. Traditional deep-learning-based methods, owing to their training and learning in a known data context and static environment, have intuitive limitations in capturing dynamic changes. We formulate the concept recommendation task as a sequential recommendation problem and model it as an MDP. Every action taken by a reinforcement learning agent relies on a constantly changing user embedding vector. Thus, HinCRec-RL can better adapt to the dynamic environment in MOOCs.

7.3 Threats to Validity

The validity of our assessment is mostly at risk from two factors.

- *Generalizability of our framework.* The main objective of our study is to improve the performance of personalized recommendations in MOOCs through a fine-grained concept recommendation approach. One threat is the differences between education scenes and other scenes (e.g. e-commerce) in the real world. In our experiments, we used a large dataset, containing 3,111,637 users and 2,527 concepts, to account for as many user behaviors as possible. Moreover, we made the training and testing datasets as significantly different as possible. In our future work, we will utilize even larger-scale datasets and datasets of other scenarios to evaluate the effectiveness of HinCRec-RL.
- *The returned recommendations may have low relevance scores.* We consider only the top 20 returned results based on the ranking score. Increasing the scale and diversity of datasets can avoid such low relevance threat. Moreover, we believe that it is reasonable to consider the top 20 results. Users typically pay attention to the top 20 or even fewer items in the recommended list in real-world recommender systems and disregard the other items. This indicates that items starting from $k + 1$ (k=20) are not significant from the perspective of users.

8 CONCLUSION

This paper presented HinCRec-RL, a novel reinforcement learning architecture that incorporates HINs, for the task of concept recommendation in MOOCs. HinCRec-RL has a better ability to learn user representations by meta-path sampling over an HIN, and can also accommodate the interactions between users and knowledge concepts in the dynamic context of MOOCs. From experimental results, we can have a conclusion that the proposed HinCRec-RL model can considerably improve both the precision and diversity of the recommended list. Our model can be applied in not only MOOC platforms for knowledge concept recommendation, but also other practical scenarios, such as user recommendation and commodity recommendation. In the future, we will deploy the concept recommendation algorithm on the real-world MOOC platform *XuetangX* and provide a commercial concepts recommendation service with personalized and dynamic characteristics. In addition, we will explore how to apply inverse reinforcement learning into the recommender system and further unleash the power of HinCRec-RL.

ACKNOWLEDGMENTS

REFERENCES

[1] Charu C Aggarwal et al. 2016. *Recommender systems*. Vol. 1. Springer.

[2] Rabeya Bosri, Mohammad Shahriar Rahman, Md Zakirul Alam Bhuiyan, and Abdullah Al Omar. 2021. Integrating blockchain with artificial intelligence for privacy-preserving recommender systems. *IEEE Transactions on Network Science and Engineering* 8, 2 (2021), 1009–1018.

[3] Xiaoyan Cai, Junwei Han, and Libin Yang. 2018. Generative adversarial network based heterogeneous bibliographic network representation for personalized citation recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[4] Yuwei Cao, Hao Peng, Jia Wu, Yingtong Dou, Jianxin Li, and Philip S. Yu. 2021. Knowledge-Preserving Incremental Social Event Detection via Heterogeneous GNNs. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 3383–3395.

[5] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Representation Learning for Attributed Multiplex Heterogeneous Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 1358–1368.

[6] Hongxu Chen, Yicong Li, Xiangguo Sun, Guandong Xu, and Hongzhi Yin. 2021. Temporal meta-path guided explainable recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*. 1056–1064.

[7] Hongxu Chen, Hongzhi Yin, Weiqing Wang, Hao Wang, Quoc Viet Hung Nguyen, and Xue Li. 2018. PME: projected metric embedding on heterogeneous networks for link prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1177–1186.

[8] Penghe Chen, Yu Lu, Vincent W Zheng, and Yang Pian. 2018. Prerequisite-driven deep knowledge tracing. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 39–48.

[9] Wei Chen, Zhendong Niu, Xiangyu Zhao, and Yi Li. 2014. A hybrid recommendation algorithm adapted in e-learning environments. *World Wide Web* 17, 2 (2014), 271–284.

[10] Xiaocong Chen, Chaoran Huang, Lina Yao, Xianzhi Wang, Wenjie Zhang, et al. 2020. Knowledge-guided deep reinforcement learning for interactive recommendation. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[11] Xinshi Chen, Shuang Li, Hui Li, Shaohua Jiang, and Le Song. 2019. Neural Model-Based Reinforcement Learning for Recommendation. In *International Conference on Learning Representations*.

[12] Zhongfen Deng, Hao Peng, Congying Xia, Jianxin Li, Lifang He, and Philip S. Yu. 2020. Hierarchical Bi-Directional Self-Attention Networks for Paper Review Rating Recommendation. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, Donia Scott, Núria Bel, and Chengqing Zong (Eds.). International Committee on Computational Linguistics, 6302–6314.

[13] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 135–144.

[14] Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and Yongliang Li. 2019. Metapath-guided heterogeneous graph neural network for intent recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2478–2486.

[15] Jun Feng, Heng Li, Minlie Huang, Shichen Liu, Wenwu Ou, Zhirong Wang, and Xiaoyan Zhu. 2018. Learning to collaborate: Multi-scenario ranking via multi-agent reinforcement learning. In *Proceedings of the 2018 World Wide Web Conference*. 1939–1948.

[16] Wenzheng Feng, Jie Tang, and Tracy Xiao Liu. 2019. Understanding dropouts in MOOCs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 517–524.

[17] Kun Gai, Xiaoqiang Zhu, Han Li, Kai Liu, and Zhe Wang. 2017. Learning piece-wise linear models from large scale data for ad click prediction. *arXiv preprint arXiv:1704.05194* (2017).

[18] Sarik Ghazarian and Mohammad Ali Nematbakhsh. 2015. Enhancing memory-based collaborative filtering for group recommender systems. *Expert systems with applications* 42, 7 (2015), 3801–3812.

[19] Jibing Gong, Shen Wang, Jinlong Wang, Wenzheng Feng, Hao Peng, Jie Tang, and Philip S Yu. 2020. Attentional graph convolutional networks for knowledge concept recommendation in moocs in a heterogeneous view. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 79–88.

[20] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 1025–1035.

[21] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. 2018. NAIS: Neural Attentive Item Similarity Model for Recommendation. *IEEE Trans. Knowl. Data Eng.* 30, 12 (2018), 2354–2366.

[22] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.

[23] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. 1–9.

[24] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Tianchi Yang. 2018. Local and Global Information Fusion for Top-N Recommendation in Heterogeneous Information Network. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1683–1686.

[25] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1531–1540.

[26] Liwei Huang, Mingsheng Fu, Fan Li, Hong Qu, Yangjun Liu, and Wenyu Chen. 2021. A deep reinforcement learning based long-term recommender system. *Knowledge-Based Systems* 213 (2021), 106706.

[27] Ran Huang and Ran Lu. 2018. Research on Content-based MOOC Recommender Model. In *2018 5th International Conference on Systems and Informatics (ICSAI)*. IEEE, 676–681.

[28] Tinglin Huang, Yuxiao Dong, Ming Ding, Zhen Yang, Wenzheng Feng, Xinyu Wang, and Jie Tang. 2021. Mixgcf: An improved training method for graph neural network-based recommender systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 665–674.

[29] Wacharawan Intayoad, Chayapol Kamyod, and Punnarumol Temdee. 2020. Reinforcement Learning Based on Contextual Bandits for Personalized Online Learning Recommendation Systems. *Wireless Personal Communications* (2020), 1–16.

[30] Jiarui Jin, Jiarui Qin, Yuchen Fang, Kounianhua Du, Weinan Zhang, Yong Yu, Zheng Zhang, and Alexander J Smola. 2020. An efficient neighborhood-based interaction model for recommendation on heterogeneous graph. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 75–84.

[31] Xia Jing and Jie Tang. 2017. Guess you like: course recommendation in MOOCs. In *Proceedings of the International Conference on Web Intelligence*. ACM, 783–789.

[32] Santosh Kabbur, Xia Ning, and George Karypis. 2013. FISM: factored item similarity models for top-N recommender systems. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*. 659–667.

[33] Hamid Karimi, Tyler Derr, Jiangtao Huang, and Jiliang Tang. 2020. Online Academic Course Performance Prediction Using Relational Graph Convolutional Neural Network. *International Educational Data Mining Society* (2020).

[34] Gang Ke, Hong-Le Du, and Yeh-Cheng Chen. 2021. Cross-platform dynamic goods recommendation system based on reinforcement learning and social networks. *Applied Soft Computing* 104 (2021), 107213.

[35] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017), 3146–3154.

[36] Asra Khalid, Karsten Lundqvist, and Anne Yates. 2022. A literature review of implemented recommendation techniques used in Massive Open online Courses. *Expert Systems with Applications* 187 (2022), 115926.

[37] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

[38] Chen Li, Linmei Hu, Chuan Shi, Guojie Song, and Yuanfu Lu. 2021. Sequence-aware Heterogeneous Graph Neural Collaborative Filtering. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM, 64–72.

[39] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*. 1419–1428.

[40] Jinzhi Liao, Jiuyang Tang, and Xiang Zhao. 2019. Course drop-out prediction on MOOC platform via clustering and tensor completion. *Tsinghua Science and Technology* 24, 4 (2019), 412–422.

[41] Feng Liu, Ruiming Tang, Xutao Li, Weinan Zhang, Yunming Ye, Haokun Chen, Huifeng Guo, and Yuzhou Zhang. 2018. Deep reinforcement learning based recommendation with explicit user-item interactions modeling. *arXiv preprint arXiv:1810.12027* (2018).

[42] Xiangyong Liu, Guojun Wang, and Md Zakirul Alam Bhuiyan. 2022. Personalised context-aware re-ranking in recommender system. *Connection Science* 34, 1 (2022), 319–338.

[43] Xiangyong Liu, Guojun Wang, and Md Zakirul Alam Bhuiyan. 2022. Re-ranking with multiple objective optimization in recommender system. *Transactions on Emerging Telecommunications Technologies* 33, 1 (2022), e4398.

[44] Babak Loni, Roberto Pagano, Martha Larson, and Alan Hanjalic. 2019. Top-N recommendation with multi-channel positive feedback using factorization machines. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 1–23.

[45] Weiming Lu, Yangfan Zhou, Jiale Yu, and Chenhao Jia. 2019. Concept Extraction and Prerequisite Relation Learning from Educational Data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 9678–9685.

[46] Shunmei Meng, Shaoyu Fan, Qianmu Li, Xinna Wang, Jing Zhang, Xiaolong Xu, Lianyong Qi, and Md Zakirul Alam Bhuiyan. 2022. Privacy-Aware Factorization-Based Hybrid Recommendation Method for Healthcare Services. *IEEE Transactions on Industrial Informatics* 18, 8 (2022), 5637–5647.

[47] Abdullah Al Omar, Rabeya Bosri, Mohammad Shahriar Rahman, Nasima Begum, and Md Zakirul Alam Bhuiyan. 2019. Towards privacy-preserving recommender system with blockchains. In *International Conference on Dependability in Sensor, Cloud, and Big Data Systems and Applications*. Springer, 106–118.

[48] Feiyang Pan, Qingpeng Cai, Pingzhong Tang, Fuzhen Zhuang, and Qing He. 2019. Policy gradients for contextual recommendations. In *The World Wide Web Conference*. 1421–1431.

[49] Liangming Pan, Chengjiang Li, Juanzi Li, and Jie Tang. 2017. Prerequisite relation learning for concepts in moocs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1447–1456.

[50] Liangming Pan, Xiaochen Wang, Chengjiang Li, Juanzi Li, and Jie Tang. 2017. Course concept extraction in MOOCs via embedding-based graph propagation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 875–884.

[51] Yanxia Pang, Yuanyuan Jin, Ying Zhang, and Tao Zhu. 2017. Collaborative filtering recommendation for MOOC application. *Computer Applications in Engineering Education* 25, 1 (2017), 120–128.

[52] Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, and Jae Kyeong Kim. 2012. A literature review and classification of recommender systems research. *Expert systems with applications* 39, 11 (2012), 10059–10072.

[53] Hao Peng, Jianxin Li, Yangqiu Song, Renyu Yang, Rajiv Ranjan, Philip S. Yu, and Lifang He. 2021. Streaming Social Event Detection and Evolution Discovery in Heterogeneous Information Networks. *ACM Trans. Knowl. Discov. Data* 15, 5, Article 89 (may 2021), 33 pages.

[54] Hao Peng, Hongfei Wang, Bowen Du, Md Zakirul Alam Bhuiyan, Hongyuan Ma, Jianwei Liu, Lihong Wang, Zeyu Yang, Linfeng Du, Senzhang Wang, and Philip S. Yu. 2020. Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. *Information Sciences* 521 (2020), 277–290.

[55] Hao Peng, Ruitong Zhang, Yingtong Dou, Renyu Yang, Jingyi Zhang, and Philip S Yu. 2021. Reinforced neighborhood selection guided multi-relational graph neural networks. *ACM Transactions on Information Systems (TOIS)* 40, 4 (2021), 1–46.

[56] Hao Peng, Ruitong Zhang, Shaoning Li, Yuwei Cao, Shirui Pan, and Philip Yu. 2022. Reinforced, incremental and cross-lingual event detection from social messages. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[57] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.

[58] Jiezhong Qiu, Jie Tang, Tracy Xiao Liu, Jie Gong, Chenhui Zhang, Qian Zhang, and Yufei Xue. 2016. Modeling and predicting learning behavior in MOOCs. In *Proceedings of the ninth ACM international conference on web search and data mining*. ACM, 93–102.

[59] Kahina Rabahallah, Latifa Mahdaoui, and Faiçal Azouaou. 2018. MOOCs Recommender System using Ontology and Memory-based Collaborative Filtering. In *ICEIS (1)*. 635–641.

[60] Steffen Rendle. 2012. Factorization Machines with libFM. *ACM TIST* 3, 3 (2012), 57:1–57:22.

[61] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback *(UAI '09)*. AUAI Press, Arlington, Virginia, USA, 452–461.

[62] Yilin Shen, Yue Deng, Avik Ray, and Hongxia Jin. 2018. Interactive recommendation via deep neural memory augmented contextual bandits. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 122–130.

[63] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and Philip S. Yu. 2019. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 31, 2 (2019), 357–370.

[64] David Silver, Leonard Newnham, David Barker, Suzanne Weller, and Jason McFall. 2013. Concurrent reinforcement learning from customer interactions. In *International conference on machine learning*. PMLR, 924–932.

[65] Changfeng Sun, Han Liu, Meng Liu, Zhaochun Ren, Tian Gan, and Liqiang Nie. 2020. LARA: Attribute-to-feature Adversarial Learning for New-item Recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 582–590.

[66] Xiangguo Sun, Hongzhi Yin, Bo Liu, Hongxu Chen, Jiuxin Cao, Yingxia Shao, and Nguyen Quoc Viet Hung. 2021. Heterogeneous hypergraph embedding for graph classification. In *Proceedings of the 14th acm international conference on web search and data mining*. 725–733.

[67] Yizhou Sun and Jiawei Han. 2012. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery* 3, 2 (2012), 1–159.

[68] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment* 4, 11 (2011), 992–1003.

[69] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*. 1057–1063.

[70] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[71] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

[72] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354.

[73] Shanshan Wan and Zhendong Niu. 2019. A hybrid e-learning recommendation approach based on learners' influence propagation. *IEEE Transactions on Knowledge and Data Engineering* 32, 5 (2019), 827–840.

[74] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous Graph Attention Network. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia (Eds.). ACM, 2022–2032.

[75] Xiao Wang, Nian Liu, Hui Han, and Chuan Shi. 2021. Self-supervised heterogeneous graph neural network with co-contrastive learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1726–1736.

[76] Max Welling and Thomas N Kipf. 2016. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*.

[77] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3-4 (1992), 229–256.

[78] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 235–244.

[79] Le Wu, Yonghui Yang, Kun Zhang, Richang Hong, Yanjie Fu, and Meng Wang. 2020. Joint item recommendation and attribute inference: An adaptive graph convolutional network approach. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 679–688.

[80] Ruobing Xie, Shaoliang Zhang, Rui Wang, Feng Xia, and Leyu Lin. 2021. Hierarchical Reinforcement Learning for Integrated Recommendation. In *Proceedings of AAAI*.

[81] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *International Conference on Learning Representations*.

[82] Hongzhi Yin, Qinyong Wang, Kai Zheng, Zhixu Li, and Xiaofang Zhou. 2020. Overcoming data sparsity in group recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[83] Jifan Yu, Gan Luo, Tong Xiao, Qingyang Zhong, Yuquan Wang, Wenzheng Feng, Junyi Luo, Chenyu Wang, Lei Hou, Juanzi Li, et al. 2020. MOOCCube: a large-scale data repository for NLP applications in MOOCs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3135–3142.

[84] Jifan Yu, Chenyu Wang, Gan Luo, Lei Hou, Juanzi Li, Jie Tang, and Zhiyuan Liu. 2019. Course Concept Expansion in MOOCs with External Knowledge and Interactive Game. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. ACL, 4292–4302.

[85] Jifan Yu, Yuquan Wang, Qingyang Zhong, Gan Luo, Yiming Mao, Kai Sun, Wenzheng Feng, Wei Xu, Shulin Cao, Kaisheng Zeng, et al. 2021. MOOCCubeX: A Large Knowledge-centered Repository for Adaptive Learning in MOOCs. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4643–4652.

[86] Runlong Yu, Qi Liu, Yuyang Ye, Mingyue Cheng, Enhong Chen, and Jianhui Ma. 2020. Collaborative list-and-pairwise filtering from implicit feedback. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[87] Tong Yu, Yilin Shen, Ruiyi Zhang, Xiangyu Zeng, and Hongxia Jin. 2019. Vision-language recommendation via attribute augmented multimodal reinforcement learning. In *Proceedings of the 27th ACM International Conference on Multimedia*. 39–47.

[88] Jing Zhang, Bowen Hao, Bo Chen, Cuiping Li, Hong Chen, and Jimeng Sund. 2019. Hierarchical Reinforcement Learning for Course Recommendation in MOOCs. *Psychology* 5, 4.64 (2019), 5–65.

[89] Sixiao Zhang, Hongxu Chen, Xiao Ming, Lizhen Cui, Hongzhi Yin, and Guandong Xu. 2021. Where Are We in Embedding Spaces? *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 2223–2231.

[90] Yutao Zhang, Robert Chen, Jie Tang, Walter F Stewart, and Jimeng Sun. 2017. LEAP: learning to prescribe effective and safe treatment combinations for multimorbidity. In *proceedings of the 23rd ACM SIGKDD international conference on knowledge Discovery and data Mining*. 1315–1324.

[91] Dongyang Zhao, Liang Zhang, Bo Zhang, Lizhou Zheng, Yongjun Bao, and Weipeng Yan. 2020. MaHRL: Multi-goals Abstraction Based Deep Hierarchical Reinforcement Learning for Recommendations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 871–880.

[92] Wei Zhao, Benyou Wang, Min Yang, Jianbo Ye, Zhou Zhao, Xiaojun Chen, and Ying Shen. 2019. Leveraging long and short-term information in content-aware movie recommendation via adversarial training. *IEEE transactions on cybernetics* 50, 11 (2019), 4680–4693.

[93] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2018. Deep reinforcement learning for page-wise recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 95–103.

[94] Xiangyu Zhao, Long Xia, Lixin Zou, Hui Liu, Dawei Yin, and Jiliang Tang. 2020. Whole-Chain Recommendations. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1883–1891.

[95] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 167–176.

[96] Sijin Zhou, Xinyi Dai, Haokun Chen, Weinan Zhang, Kan Ren, Ruiming Tang, Xiuqiang He, and Yong Yu. 2020. Interactive recommender system via knowledge graph-enhanced reinforcement learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 179–188.

[97] Yu Zhu, Jinghao Lin, Shibi He, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2019. Addressing the item cold-start problem by attribute-driven active learning. *IEEE Transactions on Knowledge and Data Engineering* 32, 4 (2019), 631–644.

[98] Lixin Zou, Long Xia, Zhuoye Ding, Jiaxing Song, Weidong Liu, and Dawei Yin. 2019. Reinforcement Learning to Optimize Long-term User Engagement in Recommender Systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2810–2818.

[99] Lixin Zou, Long Xia, Pan Du, Zhuo Zhang, Ting Bai, Weidong Liu, Jian-Yun Nie, and Dawei Yin. 2020. Pseudo Dyna-Q: A reinforcement learning framework for interactive recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 816–824.