Neural Computing and Applications (2024) 36:4115–4132
https://doi.org/10.1007/s00521-023-09257-7

ORIGINAL ARTICLE

Knowledge-aware reasoning with self-supervised reinforcement learning for explainable recommendation in MOOCs

Yuanguo Lin[1,2] · Wei Zhang[2] · Fan Lin[2] · Wenhua Zeng[2] · Xiuze Zhou[3] · Pengcheng Wu[4]

Received: 21 October 2022 / Accepted: 6 November 2023 / Published online: 10 December 2023

Abstract
Explainable recommendation is important but not yet explored in Massive Open Online Courses (MOOCs). Recently, knowledge graph (KG) has achieved great success in explainable recommendations. However, the e-learning scenario has some unique constraints, such as learners' knowledge structure and course prerequisite requirements, leading the existing KG-based recommendation methods to work poorly in MOOCs. To address these issues, we propose a novel explainable recommendation model, namely **K**nowledge-aware **R**easoning with self-supervised **R**einforcement **L**earning (KRRL). Specifically, to enhance the semantic representation and relation in the KG, a multi-level representation learning method enriches the perceptual information of semantic interactions. Afterward, a self-supervised reinforcement learning method effectively guides the path reasoning over the KG, to match the unique constraints in the e-learning scenario. We evaluate the KRRL model on two real-world MOOCs datasets. The experimental results show that KRRL evidently outperforms state-of-the-art baselines in terms of the recommendation accuracy and explainability.

Keywords Explainable recommendation · Course recommendation · Knowledge graph reasoning · Reinforcement learning

Yuanguo Lin and Wei Zhang contributed equally to this work.

✉ Fan Lin
iamafan@xmu.edu.cn

✉ Xiuze Zhou
zhouxiuze@foxmail.com

Yuanguo Lin
xdlyg@stu.xmu.edu.cn

Wei Zhang
wzhang18@stu.xmu.edu.cn

Wenhua Zeng
whzeng@xmu.edu.cn

Pengcheng Wu
pengcheng.wu@ntu.edu.sg

1 School of Computer Engineering, Jimei University, Xiamen 361021, China

2 School of Informatics, Xiamen University, Xiamen 361005, China

3 Shuye Tech, Hangzhou 310000, China

4 Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore 639798, Singapore

List of symbols

$\mathcal{G}$	A knowledge graph
(e, r, e')	A triple of head entity, relation and tail entity
$\mathcal{E}$	The entity set
$\mathcal{R}_{\mathcal{G}}$	The relation set
$u \in U$	A user ID in the user set U
$c_t \in C$	A course ID in the course corpus C
ε^u	The observed interaction of a user
c_i	A target course
$k_n \in \mathcal{K}$	A course concept in the course concept set
$\mathbf{k}_i$	The i-th concept embedding in a course
$\mathbf{w}_j$	The j-th word embedding in a concept
W_t	A weight of the edge according to (e_t, r_t, e'_t)
$\mathcal{S}$	The state set
$\mathcal{A}$	The action set
$\mathcal{P}$	The state transition probability
$\mathcal{R}$	The reward function
$s_t \in \mathcal{S}$	The state at time step t
$a_t \in \mathcal{A}$	The action at time step t
$R_{e,T}$	The reward for the path finding
$R_{p,t}$	The reward for the path discriminator
$D_p(s_t, a_t)$	A path discriminator with respect to (s_t, a_t)
P_{u,c_i}	A multi-hop path that connects u with c_i via t relations
P^D_{u,c_i}	An expert demonstration of the multi-hop path

s_t^D	The state in an expert path
a_t^D	The action in an expert path
π_θ	The actor policy
$\mathbf{a}_{\theta,t}$	The action embedding in the actor network
$\mathbf{a}_{p,t}$	The action embedding in path discriminator
Q_ϕ	The critic network
$\mathbf{a}_{\phi,t}$	The action embedding in the critic network
λ	A factor to balance $R_{e,T}$ and $R_{p,t}$
β	A decay factor of the action-value function
q_t	A target action-value in one step

1 Introduction

Currently, Massive Open Online Courses (MOOCs), a popular way of e-learning, suffer from the issue of information overload. As a notable solution arises in MOOC platforms, the recommendation technologies [1, 2] help the learners find some interesting or required courses by generating personalized recommendations. To ensure the effectiveness and improve the accuracy of course recommendations, many methods have been proposed [3, 4]. However, in reality, besides the accuracy, learners are concerned with the rationality of the recommendation process (*i.e.*, the explainability of recommendation [5, 6]), which is important but not yet explored [7].

In recent years, knowledge graph (KG) [8, 9] has been a hot topic in recommender systems [10, 11]. The experiments suggested that KGs not only improve recommendation accuracy but also enhance the explainability of the recommendation [12]. In this paper, we leverage multi-hop path reasoning over a KG to interpret the recommendation process and guide the specific learner to find feasible learning paths. Existing KG-based recommendation methods, such as policy-guided path reasoning (PGPR) [11] and ADversarial Actor-Critic (ADAC) [13] that utilize reinforcement learning (RL) [14, 15] to conduct explicit reasoning, make course recommendations poorly because the e-learning environment usually suffers from the following complex constraints.

- **Course prerequisite requirements.** In educational applications such as curriculum planning and course recommendation, prerequisite relations among courses are important [16, 17]. Course sequence recommendations should include the prerequisite courses [2], whether they are mandatory or not, as learners may lack these course concepts (*i.e.*, learning experience).
- **Learner's knowledge structure.** It is well known that learners' knowledge structure evolves in the process of learning [18]. In this case, the recommendation strategy needs to take into account learners' knowledge structure. However, in reality, it is difficult to completely construct learners' knowledge states.

To address the above challenges in course recommender systems, we propose a unified framework, namely **K**nowledge-aware **R**easoning with self-supervised **R**einforcement **L**earning (KRRL). The core idea is that the learner's knowledge structure consists of many series of knowledge points, including course concepts. Meanwhile, each course contains multiple concepts, and similar courses usually share some of the same concepts. Thus, the semantic representation of course concepts is good at capturing learners' knowledge levels. In addition, RL, with a powerful ability of self-learning from interactions [14], can automatically learn the latent relationship between users and items [19]. Based on this case, we combine the semantic perception and path reasoning over the KG, conducted by the RL agent, to improve the accuracy and explainability of course recommendation. Different from the recommendation models in other scenarios, our framework focuses on the knowledge-aware reasoning for explainable recommendation in MOOCs. It not only constructs the explicit information and implicit feedback in the learning process, but also recommends target courses that match learners' knowledge structure and course prerequisite requirements with self-supervised expert strategies. The framework has two steps:

The first step is the effective construction of courses and learners' profiles in the KG. To this end, we introduce a multi-level representation learning method to enhance the semantic representation and relation of KG. Specifically, to cast learners' sequential preferences, a course-level representation is utilized to model learners' learning behavior. Moreover, a concept-level representation is leveraged to capture learners' knowledge states as the attribute-level information of the historical courses. In particular, multiple similar courses can be associated with one or more of the same concepts. Such connectivity may reveal the potential factors in prerequisite relations among courses. For example, the course *Fundamentals of Big Data System* and its prerequisite course *Operating Systems* share some concepts, such as "process control block", "memory management", "file store", etc. This connectivity enriches the perceptual information of semantic interactions in the KG and contributes to the path reasoning of course recommendation.

The second step is the implementation of the explainable recommendation in MOOCs. It needs an efficient way to accomplish this task when labeled samples are often limited in MOOCs (*e.g.*, lack of explicit feedback). For this purpose, we propose a self-supervised RL approach to guide the path reasoning over the KG. Instead of only using the smallest m-hop relation to seek the path over an unweighted graph [11, 13], inspired by [20], we adopt a weighted operation with the similarity between entities based on their relation to distinguish the strength of different paths. The weighted action paths can adjust the policy to effectively infer learners' preferences. Moreover, an inverse reinforcement learning (IRL) algorithm [21] is introduced to find rational demonstrations. It employs expert demonstrations and reward signals to motivate the policy to enable accurate recommendations. In this way, the recommender agent can recommend courses that match learners' knowledge structure and interests, while enhancing the reasoning ability.

In summary, we have made the following major contributions:

- We present a novel framework (*i.e.*, KRRL), which conducts the knowledge-aware reasoning over the KG. To the best of our knowledge, we are the first to propose an explainable course recommendation model in MOOCs.
- We propose a multi-level representation learning method to enhance the semantic representation and relation of KG, in which course-level representation models learners' learning behavior, while concept-level representation captures learners' knowledge states as the attribute-level information of historical courses.
- To effectively recommend courses that match learners' knowledge structure and interests, we propose a self-supervised RL approach to guide the path reasoning. The self-supervised module can help the recommender agent distinguish the strength of different paths for inferring learners' preferences and find rational demonstrations to enable accurate recommendations.
- Extensive experiments on two public MOOCs datasets show that our KRRL framework not only improves the recommendation accuracy but also achieves better reasoning ability, compared with state-of-the-art baselines.

The rest of this paper is organized as follows: In Sect. 2, we review the related work. Section 3 introduces the definition of the explainable recommendation task in MOOCs. Section 4 elaborates the proposed KRRL framework. The design of the simulation experiments is presented in Sect. 5. In Sect. 6, we analyze the experimental results. Finally, Sect. 7 presents the conclusion and future work.

2 Related work

In this section, we provide a literature review of the related work in the following research areas: (1) KG-based explainable recommendation and (2) course recommender systems.

2.1 KG-based explainable recommendation

Recent advances in KG have attracted much attention in explainable recommendation [22–26]. Existing KG-based explainable recommendation methods [27, 28] can be classified into two types: embedding-based and path-based models.

2.1.1 Embedding-based models

The embedding-based models can give explanations to recommendations by learning entities and relations in the KG [29–31]. For example, [32] developed a recommendation reasoning paradigm named AnchorKG, which generates a compact anchor graph to enhance the latent representation of each news article and performs the knowledge reasoning via the interaction between different anchor graphs. [33] proposed a KG-based method for visualization recommendation, which achieves high-quality explainability without manual specifications of visualization rules. [34] developed a knowledge-enhanced recommendation model with the sequential preference representation, in which knowledge base information is incorporated into a key-value memory network to capture attribute-level user preference. The model is interpretable as it combines knowledge base information to represent users' preferences. [35] introduced self-generated and embedding-based graphs into a new graph convolution network, which can learn relationships between users or items by ever-changing multiple graphs. Moreover, [36] proposed a novel framework to provide personalized explanations based on heterogeneous knowledge base embeddings. Essentially, the above embedding-based models are post hoc explanations, since the explanations are generated by a soft matching algorithm after the target items have been recommended. However, they can hardly mine the perceptual information of semantic interactions in the KG, which is crucial to reveal the potential factors in relations among entities.

2.1.2 Path-based models

The path-based models learn the connectivity patterns between two nodes in KG to make explainable decisions [22, 37, 38]. For example, [39] employed a meta-path-based entropy encoder and recurrent neural network to improve the accuracy and explainability of recommendations. [20] proposed a new recurrent neural network (RNN) for explainable recommendation, which generates path representations over KGs to infer users' preferences. [40] adopted a hierarchical self-attention network to learn high-order semantic relevance from both entities and paths for more reasonable explanations. Most existing methods generate inaccurate explanations since they only use static KG. To address this issue, a temporal meta-path-guided mechanism [41] is proposed to model dynamic user-item evolutions on KG for better explainability. To model multi-level user preferences, [42] proposed a novel KG-based reasoning framework, in which a multi-level reasoning path extraction approach can reveal user interests.

Besides, [43] developed a user-centric path reasoning network to offer explainable recommendation, in which a multi-view structure guides the search following both sequence reasoning information and the user's demand to increase explanation diversity. [11] proposed a policy-guided path reasoning (PGPR) method by using the RL algorithm to perform the reasoning process over KGs. In particular, the PGPR method integrates a soft reward function, a multi-hop scoring approach, and a user-conditional action pruning strategy to avoid enumerating all possible paths. Furthermore, to fully explore perfect path demonstrations for improving the recommendation accuracy and explainability, an adversarial Actor-Critic framework [13] leverages path demonstrations with generative adversarial networks (GANs) to guide the path-finding process.

Most of the existing path-based recommendation models only take shorter paths between user-item pairs as better finding paths. However, they cannot distinguish the strengths of different paths, which may result in irrational explanations. In our work, we propose a self-supervised RL approach to guide the path reasoning over the KG. Instead of using shorter paths between user–item pairs to find the potential paths over an unweighted graph, we adopt a weighted operation with the similarity between entities based on their relation to distinguish the strength of different paths.

2.2 Course recommender systems

The current research on course recommender systems usually provides corresponding solutions to deal with different issues [44–48]. For example, [49] was the first to revise the user profiles by utilizing a hierarchical RL algorithm for personalized course recommendation. According to the characteristics of MOOC platforms, [50] developed a distributed computation framework based on a kind of improved apriori algorithm.

Traditional course recommendation models can be divided into four types: ontology-based method [51] makes recommendations by modeling related learners and courses. Sequence mining [18, 50] leverages the course sequence for recommending the course to a given learner. Content-based filtering [52] usually utilizes latent Dirichlet allocation (LDA) to distinguish the features of courses for recommendations. Collaborative filtering methods [53–55] adopt courses of similar features or users with similar preferences to recommend the target course. Due to the sparsity of MOOCs data and the diversification of learners' interests, these models are difficult to meet learners' individual needs. To address these challenges in course recommender systems, deep learning [56] has become a prevailing approach investigated as follows.

Graph-based methods make full use of the data structure by the entity relations in the graphs for course recommendation. For instance, [57] proposed an item-set embedding method to recommend top-*N* courses or learning paths to a specific student. In particular, they utilized a graph to learn rich latent relations among courses, in which students and courses are taken as the nodes, while enrollment relationships are taken as edges weighted by grades. [58] developed an automated construction method to model course knowledge graphs, which can be assisted in the learning path recommendation in MOOCs. Besides, a hyperedge-based graph neural network [59] was proposed to model the relationships among users, and the learned sequence-level user embedding effectively assists in MOOC recommendations. However, these graph-based methods fail to explain the course recommendation results.

Hybrid techniques have received increasing attention since course recommender systems often fall into requirements for complex scenarios [48, 60]. For example, [61] combined the collaborative filtering and content-based filtering methods according to multi-criteria for both user and course information, to recommend the target courses related to users' academic level and their preferences. [62] adopted a random walk-based neural network to capture learners' relational information and utilized a Bayesian probabilistic tensor factorization to make course recommendations. Moreover, based on the user profiles revised by hierarchical RL algorithm [49, 4] developed a dynamic attention network to track the changes of users' interests in sequential learning behaviors. Nevertheless, these hybrid techniques do not simultaneously consider course prerequisite requirements and learners' knowledge structure.

Several previous works took into account prerequisites in course recommender systems [2]. For instance, for reducing the time to graduate for students, a forward-search backward-induction method [18] combines course availability and prerequisite requirements to generate course sequences. Moreover, [16] leveraged course prerequisite relation and demographic profiles, as well as the user preference to make course recommendations by collaborative filtering method. Nevertheless, there are no solutions to the explainable recommendation in MOOCs. Our work fills the research gap by conducting knowledge-aware reasoning over a KG, thereby improving the recommendation accuracy while achieving the explainability of recommendations.

3 Problem formulation

The proposed framework combines the Actor-Critic algorithm [63] and IRL algorithm [21] to conduct the knowledge-aware reasoning for qualified candidate paths. Generally, the explainable recommendation problem in MOOCs can be formulated as follows.

Inputs. The KG is represented by $\mathcal{G} = \{(e, r, e') \mid e, e' \in \mathcal{E}, r \in \mathcal{R}_{\mathcal{G}}\}$, in which $\mathcal{E}$ is the entity set[1] and $\mathcal{R}_{\mathcal{G}}$ denotes the relation set. The triplet (e, r, e') represents that the head entity e (*e.g.*, *Data Structure II*) and the tail entity e' (*e.g.*, *Data Structure I*) are connected by the relation r (*e.g.*, *Prerequisite*). Accordingly, the inputs of the KG consist of a user set U, a course corpus C, and the observed interactions:

- Each **user** is represented by the user ID $u \in U$.
- Each **course** is denoted by the course ID $c_t \in C$. In general, each course contains some **concepts** extracted from the course concept set $\mathcal{K} = \{k_1, \cdots, k_n\}$.
- The **observed interaction** ε^u contains all the entities that the user u interacted with in the training dataset.

Outputs. Given a user u with his/her interactive data in MOOCs, our model aims to output a top-N list of courses (*i.e.*, recommended candidates) $C_i \subseteq C$ and generate corresponding reasoning path P_{u,c_i} for each target course $c_i \in C_i$. P_{u,c_i} is a multi-hop path that connects u with c_i via t relations.

The main notations and related descriptions in this paper are listed in Table 1.

4 The proposed model

Overview of KRRL Figure 1 illustrates a schematic overview of the proposed KRRL framework. In our framework, there are two following steps to conduct knowledge-aware reasoning over a KG:

In the first step, we introduce a multi-level representation learning method to enhance the semantic representation and relation of KG. More precisely, the course-level representation models learners' learning behavior by the user-course interactions. The concept-level representation captures learners' knowledge states, *i.e.*, a sequence of course concepts $\{k_1, \cdots, k_i\} \subseteq \mathcal{K}$, which are considered as the attribute-level information of historical courses. In this way, latent relations among courses can be well learned.

In the second step, to effectively guide the path reasoning over the KG, we present a self-supervised RL approach to recommend courses that match learners' knowledge structure and interests. The recommender agent starts from a learner and performs the multi-hop path reasoning over the KG and finally recommends suitable courses in the KG to the given learner. The self-supervised module in this approach contains two functions: the weighted action paths help the recommender agent distinguish the strength of different paths to infer learners' preferences. Moreover, an IRL-based path discriminator obtains rational demonstrations to enable accurate recommendations.

We utilize the Actor-Critic algorithm to train the KRRL framework. It employs the reward signals (*i.e.*, a reward $R_{e,t}$ for the path finding and the other reward $R_{p,t}$ for the path discriminator) to motivate the policy to evaluate the path reasoning for course recommendation.

In summary, the KRRL framework not only constructs the explicit information (*e.g.*, learner's learning behavior) and implicit feedback (*e.g.*, learner's knowledge level) in the learning process, but also recommends target courses with self-supervised expert strategies. In the following sections, we elaborate on the multi-level representation method used for the KG construction and the self-supervised RL approach applied to the path reasoning.

4.1 Multi-level representation learning

In course recommender systems, the effective construction of both courses and learners' profiles plays an important role in accurate recommendations. Besides, it is nontrivial to investigate prerequisite relations among courses to capture learners' knowledge levels, whereas the existing course recommendation methods usually ignore the prerequisite relations or assume that they are missing in the learner's learning process on the corresponding MOOC platform [16, 17]. However, learners may have studied the prerequisite courses offline or on other platforms. In this case, the learner's knowledge structure fails to be constructed completely.

To address this issue, we propose a multi-level representation learning method to enhance the semantic representation and relation of KG, according to the semantic space of different levels (*i.e.*, course level and concept level). Specifically, to capture learners' sequential preferences for courses, we utilize a course-level representation to model learners' learning behavior from the interaction sequence. In addition, we leverage a concept-level representation to grasp learners' knowledge states as the

[1] Note that the entity e can be the representation of each course, concept, category, or learner in this paper.

Table 1 Statistics of two MOOC datasets

Datasets	Users	Interactions	Courses	categories	Concepts	Prerequisites
MOOCCourse	82,535	458,453	1302	23	27,173	411
MOOCCube	55,203	354,541	706	20	23,207	352

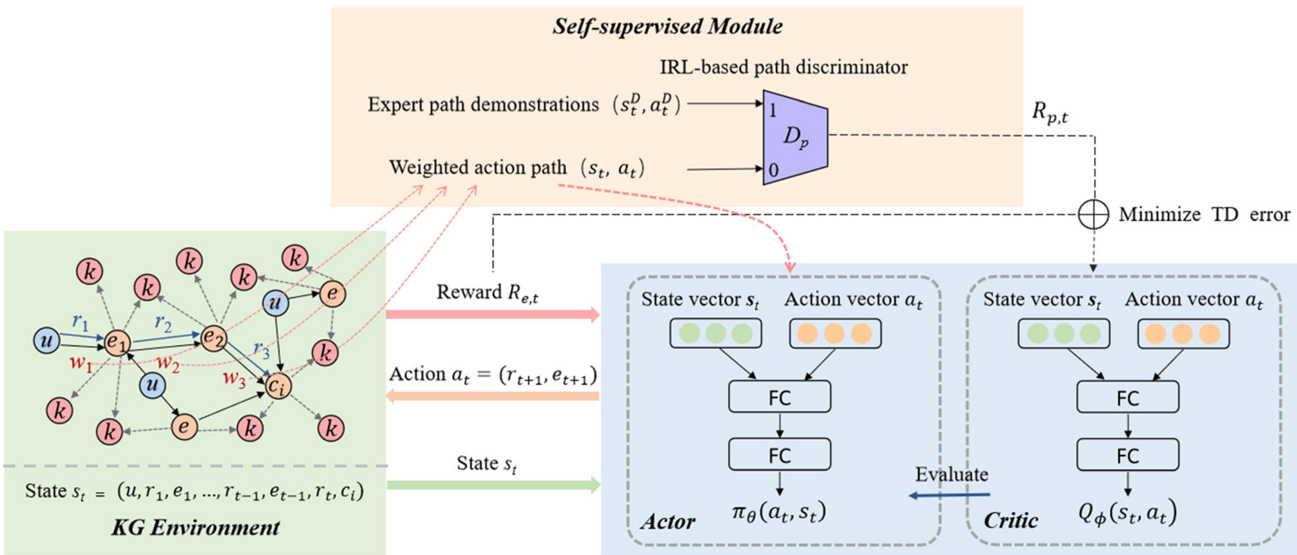

Fig. 1 The overview of our KRRL framework. The recommender agent conducts knowledge-aware reasoning to seek feasible candidate paths for recommendations by interacting with the KG environment

attribute-level information of historical courses. In this way, our method integrates the sequential preference with the attribute-level preference to better model the learner's profile, while mining the potential factors in prerequisite relations among courses since similar courses usually share some of the same concepts.

For a fair comparison, the information of KG embedding is encoded via KG embedding techniques [36, 64], which is also utilized in PGPR [11] and ADAC [13]. Specifically, it learns a distributed vector $\mathbf{e}_i \in \mathbb{R}^{d_E}$ for entity e_i and a vector $\mathbf{r} \in \mathbb{R}^{d_E}$ for relation r, where d_E is the dimension of the entity embedding. In this way, the learned embeddings offer a general representation for entities and the corresponding relations [34], which is conducive to the subsequent task (*i.e.*, the path reasoning).

4.1.1 Course-level representation method

In the KG for course recommendation, let U denote a set of learners and C denote a set of courses, given an observed interaction ε^u, our task aims to seek a recommendation path of the corresponding target course $c_i \in C_i$ for a specific learner $u \in U$. To this end, by sorting the interaction records in time sequence, the interaction sequence of the learner u can be formed as $\{c_1^u, \cdots, c_t^u, \cdots, c_{t_u}^u\}$, where c_t^u denotes the historical course $c \in C$ enrolled by the learner u at time t, and t_u is the number of historical courses that the learner enrolled in. Accordingly, the course-level representation method is used to encode the courses. That is, the embedding vector $\mathbf{c}_t^u \in \mathbb{R}^{d_E}$ for the historical course c_t^u is called historical course embedding, and the embedding vector $\mathbf{c}_i \in \mathbb{R}^{d_E}$ for the target course c_i is called target course embedding. Thus, to capture the learner's sequential preference for courses, we can use the course-level representation to model the learner's learning behavior by an interaction sequence of the learner.

Since the course-level representation method is not competent to understand and explain the hidden vector of each course, it is difficult to learn the learner's knowledge states from the interaction sequence. Hence, we propose the following concept-level representation to address this challenge.

4.1.2 Concept-level representation method

It is widely known that the learner's knowledge structure consists of many series of knowledge points, including course concepts (*i.e.*, knowledge concepts) [65]. Besides, each course contains multiple concepts, and there are often the same concepts between similar courses. In this case, we can leverage the semantic representation of course concepts to capture the learner's knowledge structure.

More precisely, the concept-level representation method can capture the learner's knowledge states by a sequence of course concepts, *i.e.*, $\{\mathbf{k}_1, \ldots, \mathbf{k}_i\} \subseteq \mathcal{K}$, where $\mathbf{k}_i$ denotes the embedding vector of the course concept in a historical course. It can be regarded as the attribute-level information of the historical course. In general, the course concept embedding consists of a sequence of word vectors. Formally, the course embedding can be structured by a set of vector pairs according to a series of concept embeddings:

$$\mathbf{c}_t = \{(\mathbf{k}, \mathbf{w}) \mid (\mathbf{k}_i, \mathbf{w}_j), n > i > 0, j > 0\}, \tag{1}$$

where $\mathbf{k}_i$ denotes the embedding vector of the i-th concept in the course embedding $\mathbf{c}_t$, n is the number of the course concepts, and $\mathbf{w}_j$ denotes the embedding vector of the j-th word in the concept embedding $\mathbf{k}_i$.

Especially, similar to the mixed concept mapping method [66], multiple similar courses can be associated with one or more of the same concepts. Such connectivity may reveal the potential factors in prerequisite relations among courses. For example, the course *Genetics* and its prerequisite course *Cell Biology* share some concepts, such as "gene" and "cell". In this case, this semantic perception matches the learner's knowledge level and interest if the learner enrolled in the course *Genetics* or *Cell Biology*. Hence, our method enriches the perceptual information of semantic interactions in the KG, which contributes to the construction of the KG environment.

4.2 Self-supervised RL for reasoning

The second step in our framework is making explainable recommendations for the learners. Since labeled samples are often limited in MOOCs (*e.g.*, lack of explicit feedback), it requires an efficient way to accomplish this task. To this end, we propose a self-supervised RL approach to guide the path reasoning over the KG represented by the multi-level representation method. Specifically, starting from a given learner in the observed interaction ε^u, the recommender agent performs the multi-hop path reasoning over the KG, thereby recommending suitable courses that not only are desirable (*i.e.*, personalization with learner's knowledge structure), but also satisfy constraints (*e.g.*, prerequisite requirements). The self-supervised module in this approach helps the recommender agent distinguish the strength of different paths to infer learners' preferences and find rational demonstrations to achieve accurate recommendations.

4.2.1 Markov decision process

We formulate the path reasoning problem as a Markov decision process (MDP) [14]. The agent tries to recommend suitable courses for a given learner by performing the multi-hop path reasoning over the KG. Formally, the MDP can be defined by a 4-tuple $<\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}>$, where $\mathcal{S}$ denotes a set of states, $\mathcal{A}$ refers to a set of actions, $\mathcal{P}$ is a state transition probability as $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$, and $\mathcal{R}$ is the reward function.

State. The state $s_t \in \mathcal{S}$ denotes the seek status of the agent at time step t in the KG. Here, it is assumed that the path-finding process encodes a m-hop relation among a given learner u and each target course c_i, *i.e.*, the initial state $s_0 = u$, and $s_t = (u, r_1, e_1, \cdots, r_{t-1}, e_{t-1}, r_t, c_i)$. To enhance the reasoning ability of the agent for better recommendation accuracy, we introduce course concepts as auxiliary information to increase path connectivity.

Action. According to the state s_t, the agent conducts the action $a_t = (r_{t+1}, e_{t+1})$ following the policy to predict the feasible outgoing edges of entity e_t except for the history entities. It is necessary to control the size of the action space since some entities have large out-degrees in the KG. Thus, we utilize the weighted actions to retain the promising edges, which adjusts the policy to infer learners' preferences. It will be described in more detail in the next subsection. Formally, the action space $\mathcal{A}_t$ can be defined by:

$$\mathcal{A}_t = \{(r, e) \mid (e, r, e') \in \mathcal{G}, e' \notin \mathcal{E}_T, r \in \mathcal{R}_{\mathcal{G}}\}, \tag{2}$$

where $\mathcal{E}_T$ denotes the historical entity set.

Transition. In the KG, any state except for the initial state is determined by history entity and relation. For all $s, s', s_t \in \mathcal{S}$, and $a, a_t \in \mathcal{A}$, the transition probability to the next state is deterministic [32]:

$$\mathcal{P}(s' \mid s, a) \doteq Pr\{s_{t+1} = s' \mid s_t = s, a_t = (r_{t+1}, e_{t+1})\} = 1, \tag{3}$$

where s_{t+1} denotes the state at time $t+1$.

Reward. The reward function $\mathcal{R}$ is the terminal reward, which measures whether the agent generates a m-hop path that starts from a given learner u and ends with a target course c_i. Formally, the reward for the path-finding $R_{e,T}$ at the final time step T can be defined as follows:

$$R_{e,T} = \mathbb{I}_{P(u,e_T)} = \begin{cases} 1, & \text{if } e_T \in C_i; \\ 0, & \text{if } e_T \notin C_i, \end{cases} \tag{4}$$

where $\mathbb{I}_{P(u,e_T)}$ denotes the indicator function of the path finding [13], *i.e.*, it is 1 when $e_T \in C_i$, whereas it is 0 when $e_T \notin C_i$.

4.2.2 The self-supervised module

As mentioned earlier, the self-supervised module in our KRRL framework contains two functions: One is the weighted action path, which helps the recommender agent distinguish the strength of different paths to infer learners' preferences. The other is the IRL-based path discriminator, which can obtain rational demonstrations to enable accurate recommendations. The implementation details of the two functions are elaborated as follows.

Weighted action path. Some studies assume that shorter paths are more explainable for recommendations and then adopt the smallest m-hop relation to reason the path over an unweighted graph [11, 13]. However, we argue that this method does not fully explore dependencies between entities and holistic semantics of paths, which may lead to irrational reasoning. As an alternative method similar to [20], the weighted operation with the similarity between entities based on their relation can learn dependencies between entities and distinguish the strength of different paths. Given any triplet (e_t, r_t, e'_t) that represents the head entity e_t and the tail entity e'_t are connected by the relation r_t, the weight of each edge in the path can be defined as follows:

$$W_t(e_t, r_t, e'_t) = \|\boldsymbol{V} - \boldsymbol{V}'\|^2, \boldsymbol{V} = \mathbf{e}_t + \mathbf{r}_t, \boldsymbol{V}' = \mathbf{e}'_t, \tag{5}$$

where $W_t(e_t, r_t, e'_t)$ denotes a weight of the edge with respect to the triplet (e_t, r_t, e'_t), $\boldsymbol{V}$ is the vector that represents the sum of a vector of the head entity $\mathbf{e}_t$ and a vector of its relation $\mathbf{r}_t$ with a vector of the tail entity $\mathbf{e}'_t$, and $\boldsymbol{V}'$ is the vector of the tail entity $\mathbf{e}'_t$. When the weight value of each edge in the path is smaller, the dependency between two entities on the path is stronger, because the two entities are closer in the vector space.

Based on the weighted action path, KRRL uses the Dijkstra algorithm [67] to generate the shortest path between a given learner u and a target course c_i over the weighted graph. The shortest path can be considered as a demonstration. Thus, KRRL repeats this process for all $u \in U$ and $c_i \in C$ to obtain a set of expert demonstrations P^D_{u,c_i}:

$$P^D_{u,c_i} = \left[u \overset{\min W_1(u,r_1,e_1)}{\longrightarrow} e_1 \overset{\min W_2(e_1,r_2,e_2)}{\longrightarrow} e_2 \ldots \overset{\min W_t(e_t,r_t,c_i)}{\longrightarrow} c_i\right], \tag{6}$$

where $\min W_t(e_t, r_t, c_i)$ denotes the minimum weight of the edge with respect to the triplet (e_t, r_t, c_i). In this way, the recommender agent leverages the weighted action paths to adjust the policy for inferring learners' preferences effectively, since the path weights can explore holistic semantics of paths in the observed interactions.

IRL-based path discriminator. Inspired by the ADAC model, we adopt generative adversarial imitation learning [21], a promising IRL algorithm, to obtain rational demonstrations that conform to pre-defined meta-paths. It employs expert demonstrations and reward signals to motivate the policy to enable accurate recommendations. In this way, the recommender agent can recommend courses that match learners' knowledge structure and interests, while enhancing the reasoning ability.

Specifically, the actor cooperates with a path discriminator D_p in an adversarial way: The actor first generates the paths, then the path discriminator distinguishes expert demonstrations from the paths, while the actor attempts to fool the path discriminator by imitating the expert demonstrations. Formally, the path discriminator $D_p(s_t, a_t)$ with respect to the action a_t in state s_t at time t can be formulated as follows [13].

$$\begin{aligned} &\Phi_p = tanh(\mathbf{s}_t \oplus \mathbf{a}_{p,t}), \\ &D_p(s_t, a_t) = \sigma\left(\xi_p^T tanh(W_p \Phi_p)\right), \end{aligned} \tag{7}$$

where $\mathbf{s}_t \in \mathbb{R}^{d_s}$ is the embedding vector of the state s_t, $\mathbf{a}_{p,t} \in \mathbb{R}^{d_d}$ is the embedding vector of the action $a_{p,t}$ in the path discriminator D_p, $tanh(\cdot)$ is the hyperbolic tangent function, $\sigma(\cdot)$ denotes the logistic sigmoid function, $\xi_p \in \mathbb{R}^{d_a}$ and $W_p \in \mathbb{R}^{d_a*(d_s+d_d)}$ are the parameters to be learned with d_a as the dimension of the action embedding in the actor network, d_s as the dimension of the state embedding, and d_d as the dimension of the action embedding in the path discriminator.

The discriminator is trained to calculate the probability $D_p(s_t, a_t)$ that the pair $((s_t, a_t))$ comes from the observed demonstrations. Generally, it is can be achieved by minimizing the following classification loss $\mathcal{L}_\eta$:

$$\mathcal{L}_\eta = -\left(logD_p(s^D_t, a^D_t) + log(1 - D_p(s_t, a_t))\right), \tag{8}$$

where the state $s^D_t = (u, r^D_1, e^D_1, \cdots, r^D_{t-1}, e^D_{t-1}, r^D_t, c^D_i)$ and the action $a^D_t = (r^D_{t+1}, e^D_{t+1})$ are determined by an expert path, which is randomly sampled from the observed demonstrations P^D_{u,c_i}.

When the actor generates the pair (s_t, a_t) similar to that of the observed demonstrations, it can obtain the reward $R_{p,t}$ for the path discriminator as follows [13]:

$$R_{p,t} = logD_p(s_t, a_t) - log(1 - D_p(s_t, a_t)) \tag{9}$$

To smoothly update the policy to find the promising paths approximated the observed demonstrations, we define an aggregated reward R_t by a linear combination of the rewards for the path finding and the path discriminator.

$$R_t = \lambda R_{e,T} + (1 - \lambda) R_{p,t}, \tag{10}$$

where $\lambda \in [0, 1]$ is a factor to balance the reward $R_{e,T}$ for the path finding and the reward $R_{p,t}$ for the path discriminator.

4.3 Optimization

Motivated by [13], to better guide the path reasoning and estimate the action-value, we adopt the Actor-Critic algorithm [63] to train our KRRL framework. The actor network learns the path reasoning policy according to the value function from the critic, and the critic network utilizes the temporal difference method [68] to update the action-value function in one step.

Actor. The actor network aims to learn a path reasoning policy by calculating the probability distribution of each action $a_t \in \mathcal{A}_t$ in state s_t. It leverages both the weighted action paths and expert path discriminators to effectively guide the path reasoning. We train the actor network $\pi_\theta(a_t, s_t)$ with a fully connected neural network of multiple layers [13]:

$$h_\theta = \text{ReLU}\ (W_{\theta,s}\mathbf{s}_t),$$
$$\pi_\theta(a_t, s_t) = \frac{\mathbf{a}_{\theta,t}\ \text{ReLU}\ (W_{\theta,a}h_\theta)}{\sum_{a_i\in\mathcal{A}_t} \mathbf{a}_i\ \text{ReLU}\ (W_{\theta,a}h_\theta)}, \tag{11}$$

where $ReLU(\cdot)$ is served as the activation function, $\mathbf{a}_{\theta,t} \in \mathbb{R}^{d_a}$ is the embedding vector of the action a_t in the actor network, $W_{\theta,s} \in \mathbb{R}^{d_h * d_s}$ and $W_{\theta,a} \in \mathbb{R}^{d_a * d_h}$ are the parameters of the actor network to be learned with d_h as the dimension of the hidden layer, d_s as the dimension of the state embedding, and d_a as the dimension of the action embedding.

Here, the actor network is optimized by the Policy Gradient method [69]. For each sampled trajectory, the gradients of $J_{actor}(\theta)$ can be computed by:

$$\nabla J_{actor}(\theta) \propto Q_\phi(s_t, a_t)\nabla_\theta log\pi_\theta(a_t, s_t), \tag{12}$$

where the symbol $\propto$ denotes "proportional to", and $Q_\phi(s_t, a_t)$ is the action-value function of the action a_t in state s_t. Thus, we can learn the actor by minimizing the loss function as follows [13]:

$$\mathcal{L}_{actor}(\theta) = -\mathbb{E}_{a\sim\pi_\theta}\left[Q_\phi(s_t, a)\right], \tag{13}$$

where $\mathbb{E}_{a\sim\pi_\theta}[\cdot]$ denotes the expected value of a variable given by following the actor policy π_θ.

Critic. To accurately evaluate the contribution of each action in the MDP environment, the critic network [70] is used to estimate the action-value function. It can model the rewards for both the path finding and path discriminator to guide the actor effectively. The critic network Q_ϕ calculates the action-value in state s_t:

$$h_\phi = \text{ReLU}\ (W_{\phi,s}\mathbf{s}_t),$$
$$Q_\phi(s_t, a_t) = \mathbf{a}_{\phi,t}\ \text{ReLU}\ (W_{\phi,a}h_\phi), \tag{14}$$

where $\mathbf{a}_{\phi,t} \in \mathbb{R}^{d_a}$ is the embedding vector of the action a_t in the critic network, $W_{\phi,s} \in \mathbb{R}^{d_h * d_s}$ and $W_{\phi,a} \in \mathbb{R}^{d_a * d_h}$ are the parameters of the critic network to be learned [13].

The critic network is trained by the temporal difference method, which updates a target q_t in one step according to the Bellman equation [71] as follows:

$$q_t = R_t + \mathbb{E}_{a\sim\pi_\theta}\left[\beta Q_\phi(s_{t+1}, a)\right], \tag{15}$$

where $\beta \in [0, 1]$ is a decay factor of the action-value function $Q_\phi(s_{t+1}, a)$. Thus, the critic can be learned by minimizing the temporal difference error [14]:

$$\mathcal{L}_{critic}(\phi) = \left(Q_\phi(s_t, a_t) - q_t\right)^2 \tag{16}$$

We jointly optimize the IRL-based path discriminator $D_p(s_t, a_t)$, actor network π_θ, and critic network Q_ϕ for the KRRL framework by minimizing the total loss. As such, the objective function of KRRL can be defined by:

$$\mathcal{L}_{total} = \mathcal{L}_\eta + \mathcal{L}_{actor}(\theta) + \mathcal{L}_{critic}(\phi) \tag{17}$$

5 Experimental settings

In this section, we provide the description of datasets, the baseline methods, evaluation metrics, and implementation details of the proposed model. The experiments are designed to answer our research questions as follows:

- *RQ1* Does our proposed model outperform the state-of-the-art baselines for course recommendation? How much is the improvement in terms of recommendation accuracy?
- *RQ2* Do the multi-level representation method and self-supervised RL approach improve the performance of KRRL? Which major factors affect the effectiveness of KRRL?
- *RQ3* How is the recommendation performance of KRRL when using different sampling sizes in path reasoning? Which sample size can show better performance?
- *RQ4* How is the explainability of different path reasoning methods? Which components contribute the most to the explainability of KRRL?
- *RQ5* Does our proposed model actually interpret the process of course recommendation?

5.1 Datasets

The experiments are conducted on two real-world datasets, *i.e.*, MOOCCourse[2] and MOOCCube collected from

[2] http://moocdata.cn/data/course-recommendation.

XuetangX.[3] The MOOCCourse dataset consists of 1,302 courses from 23 categories, 27,173 concepts, 411 prerequisites, 82,535 users who enrolled in more than 2 courses, and 458,453 user-course interactions chosen from October 1, 2016, to March 31, 2018. The MOOCCube dataset consists of 706 courses from 20 categories, 23,207 concepts, 352 prerequisites, 55,203 users who enrolled in more than 3 courses, and 354,541 user-course interactions chosen from June 23, 2015, to November 13, 2019. Each dataset contains user IDs, course IDs, course names, categories, course concepts, and prerequisite relations. Based on the information for each dataset, we can construct the knowledge graph that contains 4 types of entities (*i.e.*, *user*, *course*, *course concept*, and *category*) and 6 types of relations (*i.e.*, $course \xrightarrow{Title} course\ name$, $course \xrightarrow{Described\_by} course\ concept$, $course \xrightarrow{Belong\_to} category$, $course \xrightarrow{Prerequisite} course$, $user \xrightarrow{Enroll\_in} course$, as well as $user \xrightarrow{Enroll\_together} course$). We randomly sample 70% of the user–item interactions as the training set and take the rest 30% as the test set. Details about the two datasets are shown in Table 1.

5.2 Baseline methods

We compare our framework with the following competitive recommendation models.

- **BPR** [72]: It is a pairwise ranking method to learn latent embeddings of learners and courses for the top-*N* recommendations.
- **BPR-HFT** [73]: It is a model that contains the hidden factors and topics (HFT) based on topic distributions to learn latent factors.
- **LightGCN** [74]: This recommendation model learns the course and student embeddings by linearly propagating them on the student-course interaction graph.
- **RuleRec** [37]: It adopts the KG to construct a rule-guided model to make recommendations with the induced rules.
- **TP-GNN** [75]: This personalized recommendation model leverages graph neural network (GNN) and the attention mechanism to make top-*N* recommendations in MOOCs.
- **PGPR** [11]: This framework employs a policy-guided method to perform the reasoning process over KGs.
- **ADAC** [13]: It leverages an adversarial Actor-Critic method to guide the path-finding process over the KG.

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

5.3 Evaluation metrics

We evaluate the recommendation performance in terms of four widely-used metrics, *i.e.*, Recall, Precision, Hit Ratio (HR), and Normalized Discounted Cumulative Gain (NDCG). All the metrics are calculated according to the top-10 recommended courses for every learner in the test set.

The explainability of recommendations can be calculated by the Explainability Recall and Explainability Precision [76, 77]. Similar to [13], we adopt two evaluation criteria by leveraging the course concepts of each historical course. The basic idea is that a series of these course concepts can form the ground-truth information, which reveals the potential reason for recommending the target course that matches learners' knowledge structure and interests. Thus, there is good explainability if a reasoning path includes some entities mentioned in the ground-truth information. In this way, the explainability can be evaluated by matching the entities in the reasoning path with the ground-truth words. Moreover, entities whose types are *course*, *course concept*, or *category* are all mapped into the ground-truth words with the string matching.

5.4 Implementation details

To fairly compare KRRL with baselines, our MDP environment is implemented mainly according to [11] and [13]. Specifically, for the KG environment, the dimension of the entity embedding d_E is set to 100, the maximum length of the reasoning path is set to 3. For the self-supervised RL module, the maximum size of the pruned action space is set to 250, the reward weight λ is set to 0.008, the dimension of the hidden layer d_h is set to 512, the dimension of the state embedding d_s is set to 400, the dimension of the action embedding d_a is set to 256, and the dimension of the action embedding in the path discriminator d_d is set to 256. In the training process, all the parameters of the neural networks are initialized with the Adam optimization, in which the learning rate is set to 0.0001 and the batch size is set to 512 for both datasets. In the path-reasoning process for both datasets, the sampling size at different steps (*i.e.*, m_1, m_2, m_3) is set to 25, 5, 1, respectively.

6 Results and discussion

In this section, we discuss the experimental results, including the recommendation comparison, the quality of path reasoning, and the influence of sampling sizes on the recommendation performance. We also conduct an ablation analysis and a case study.

Table 2 The recommendation accuracy of several comparison methods on two MOOC datasets in terms of Recall, Precision, HR, and NDCG (%)

Methods	MOOCCourse				MOOCCube			
	Recall	Precision	HR	NDCG	Recall	Precision	HR	NDCG
BPR	6.826	0.910	14.512	4.375	5.132	1.160	9.517	3.013
BPR-HFT	7.013	1.105	15.600	5.289	6.244	1.256	9.865	3.822
LightGCN	9.025	1.396	17.113	6.980	7.251	1.300	10.921	5.085
RuleRec	9.632	1.661	17.949	7.357	7.670	1.411	11.386	5.891
TP-GNN	12.578	1.898	19.244	8.530	9.491	1.757	15.635	6.602
PGPR	16.126	2.165	22.887	9.026	11.632	2.008	20.029	7.964
ADAC	18.310	2.507	23.913	10.269	12.016	2.292	20.817	8.336
KRRL-D	18.293	2.469	23.921	10.196	12.010	2.255	20.901	8.325
KRRL-M	18.449	2.739	24.002	10.282	12.126	2.385	21.198	8.713
KRRL-W	18.752	2.741	24.127	10.318	12.294	2.342	21.276	8.790
KRRL-P	20.847	2.912	25.765	11.643	12.537	2.424	21.408	9.150
KRRL	**21.350**	**3.032**	**26.307**	**12.141**	**13.079**	**2.460**	**21.732**	**9.407**
Imp	+16.6	+20.9	+10.0	+18.2	+8.8	+7.3	+4.4	+12.8

The best results are highlighted in bold

6.1 Recommendation accuracy (RQ1)

Table 2 reports the performance comparison of different course recommendation models. KRRL framework achieves the best performance on both MOOCCourse and MOOCCube datasets in terms of different evaluation metrics. From the results in Table 2, we have the following discussion.

- It is clear that our KRRL consistently outperforms all other competitive models on both MOOCCourse and MOOCCube datasets in terms of recall, precision, HR, and NDCG. For example, compared with the ADAC model, our KRRL obtains remarkable improvements by 16.6% in terms of Recall@10, 20.9% in terms of Precision@10, 10.0% in terms of HR@10, and 18.2% in terms of NDCG@10 on the MOOCCourse dataset, while it improves the recommendation accuracy by 8.8% in terms of Recall@10, 7.3% in terms of Precision@10, 4.4% in terms of HR@10, and 12.8% in terms of NDCG@10 on the MOOCCube dataset. These well demonstrate the effectiveness of our proposed framework.
- It can be seen that traditional recommendation models (*i.e.*, BPR, BPR-HFT) are evidently outperformed by our KRRL. One possible reason is that they fail to effectively form learners' profiles for making course recommendations. Besides, the results of LightGCN are slightly worse than RuleRec, and both of them perform worse than TP-GNN on both datasets. The situation shows that the introduction of more influence factors may be helpful in improving the recommendation strategies for graph-based models, since TP-GNN leverages GNN and the attention mechanism to make recommendations.
- We also notice that PGPR and ADAC are the state-of-the-art baselines for course recommendations. This is because they are the path-based models that leverage the agent to conduct the path reasoning over KGs. Meanwhile, RuleRec is obviously worse than PGPR and ADAC since RuleRec only adopts KGs to make course recommendations with the induced rules. It indicates that the combination of RL and KG technology can provide substantial benefits for course recommendation.
- Among the KG-based models, it is clear that KRRL significantly outperforms RuleRec, PGPR, and ADAC on both datasets. The main reason is that our KRRL leverages the multi-level representation and self-supervised RL approach to enhance the ability of path reasoning. Thus, it can adjust the recommendation strategies to make more accurate recommendations. The results of KRRL are significantly better than RuleRec. It demonstrates again the superiority of path reasoning with RL and KG for course recommendation.
- All the comparison methods achieve better performances on MOOCCourse than those on MOOCCube in most cases. The potential reason is that the MOOCCourse dataset contains many more entities and relations than those from the MOOCCube dataset, which helps to construct richer associations to recommend target courses. It suggests that we can enrich the KG with more entities and relations to assist the recommendation strategies, which contributes to the recommendation accuracy, and thereby helps learners improve their learning efficiency.

6.2 Ablation analysis (RQ2)

To investigate the significance of key components in KRRL, we study how they affect the performance by comparing the following four variants of KRRL.

- **KRRL-D** is the simplified version of KRRL that does not take into account the IRL-based path discriminator.
- **KRRL-M** is the simplified version of KRRL that does not introduce the multi-level representation method, *i.e.*, ignoring the concept-level representation.
- **KRRL-W** is the simplified version of KRRL that ignores the weighted operation to distinguish the strength of different paths.
- **KRRL-P** is the simplified version of KRRL that does not introduce the course prerequisite relation into the meta-paths.

From the results shown in Table 2, we have the following observations:

- We can clearly observe that ignoring any one of all key components will result in a performance drop. For example, compared with KRRL-D, KRRL-M, KRRL-W, and KRRL-P, KRRL improves 16.7%, 15.7%, 13.8%, and 2.4% in terms of Recall@10 on MOOCCourse, and 8.9%, 7.8%, 6.4%, and 4.3% on MOOCCube. Besides, KRRL achieves an improvement of 22.8%, 10.7%, 10.6%, and 4.1% over KRRL-D, KRRL-M, KRRL-W, and KRRL-P in terms of Precision@10 on MOOCCourse, and 9.1%, 3.1%, 5.0%, and 1.5% on MOOCCube. These results well demonstrate the superiority of the KRRL framework and the effectiveness of these key components.
- KRRL-D performs the worst among these variants of KRRL on both datasets, in terms of different evaluation metrics. This demonstrates that the IRL-based path discriminator can leverage expert demonstrations and reward signals to motivate the policy to enable accurate recommendations. It also proves the indispensability of the IRL-based path discriminator in our framework, that is, without the consideration of the IRL-based path discriminator, KRRL may generate a sub-optimal recommendation result.
- The recommendation performance of KRRL-M is worse than both KRRL and KRRL-P on MOOCCourse, and it even performs worse than KRRL, KRRL-P, and KRRL-W on MOOCCube in most cases. The results verify the benefits of using the multi-level representation method to enhance the semantic representation and relation of KG. Thus, our framework effectively captures learners' knowledge levels and then generates more accurate recommendations for the learners.
- Both KRRL and KRRL-P perform better than KRRL-W on both datasets. The comparisons show the superiority of supplementing the task of course recommendation with the weighted action paths, which distinguish the strength of different paths to infer learners' preferences over the KG. Consequently, the agent can be well guided to conduct the path reasoning over the KG, thereby finding the target courses that meet learners' preferences.
- KRRL-P is outperformed by KRRL, although it achieves better performance than other variants of KRRL. It indicates that incorporating the course prerequisite relations can be conducive to improving the recommendation strategies. This demonstrates that course prerequisite requirements are important in the course sequence recommendation, since learners often consider these requirements in their curriculum planning and learning goals.

6.3 Sampling size in path reasoning (RQ3)

In this part, we study how is the recommendation performance of KRRL when using different sampling sizes in path reasoning. Table 3 reports the influence of different sampling sizes in path reasoning on the recommendation performance on both datasets. There are 8 different combinations of sampling sizes designed, and each tuple (m_1, m_2, m_3) denotes that the top m_t actions at the t-th step are sampled.

As shown in Table 3, in the combinations of (20, 3, 2), (10, 4, 3), (15, 5, 2), and (10, 5, 3), KRRL performs better than that in other combinations in most cases. For example, on the MOOCCourse dataset, KRRL in the combination of (10, 5, 3) has the best performance in terms of Recall, Precision, HR, and NDCG. On the MOOCCube dataset, KRRL in the combination of (15, 5, 2) performs the best in terms of Recall; KRRL in the combination of (20, 3, 2) has the best performance in terms of Precision and HR; and KRRL in the combination of (20, 3, 2) performs the best in terms of NDCG. These results indicate that the recommendation performance of KRRL can be improved when the sample size at the last step is large. The reason owes to more recommended options if the sample size at the last step is larger, which is conducive to selecting the optimal action of path reasoning. To further verify this conclusion for a fair comparison, the total number of sampling paths (i.e., $m_1 * m_2 * m_3$) is fixed to 120, such as (10, 12, 1) , (20, 3, 2), and (10, 4, 3). As shown in Table 3, in the combinations of (20, 3, 2) and (10, 4, 3), KRRL significantly outperforms that in the combination of (10, 12, 1), in terms of Recall, Precision, HR, and NDCG on both datasets.

Table 3 The recommendation accuracy with different sampling sizes in path reasoning on two MOOC datasets in terms of Recall, Precision, HR, and NDCG (%)

Sizes	MOOCCourse				MOOCCube			
	Recall	Precision	HR	NDCG	Recall	Precision	HR	NDCG
25, 5, 1	21.350	3.032	26.307	12.141	13.079	2.460	21.732	9.407
30, 6, 1	21.935	3.053	26.665	12.354	13.175	2.442	21.619	9.386
20, 7, 1	21.837	3.016	26.390	12.323	13.050	2.389	21.182	9.291
15, 5, 2	25.280	3.709	29.322	15.640	**13.182**	2.464	21.738	9.410
10, 5, 3	**27.012**	**3.878**	**30.811**	**16.479**	13.109	2.461	21.693	9.420
10, 4, 3	26.813	3.872	30.712	16.435	13.061	2.469	21.745	**9.468**
20, 3, 2	24.106	3.667	28.475	15.282	13.015	**2.503**	**21.954**	9.332
10, 12, 1	21.908	2.965	26.184	12.183	12.831	2.328	20.646	9.077

In addition, it is clear that the total number of the sample sizes at the first two steps, *i.e.*, $m_1 * m_2$, plays an important role in path reasoning. For example, on the MOOCCourse dataset, in the combinations of (10, 12, 1), (25, 5, 1), (20, 7, 1), and (30, 6, 1), the recommendation accuracy of KRRL is gradually improved. There is a similar trend on the MOOCCourse dataset, except for the combination of (25, 5, 1). The potential reason is that the policy tends to converge to selecting the optimal action if the sample sizes at the first two steps are large [11]. It is also noticed that the comparisons of (15, 5, 2) and (20, 3, 2), (10, 5, 3) and (10, 4, 3) show similar trends in most cases.

Generally, the sampling size in path reasoning affects the recommendation performance of KRRL to some extent. Specifically, KRRL has more excellent performance when the sample size at the last step is larger and performs better in increasing the total number of the sample sizes at the first two steps.

6.4 Quantitative analysis of explainability (RQ4)

One of the common measurements of explainability is to evaluate the percentage of recommendations that can be explained by the recommendation model [76]. Following [78], we measure the explainability of the reasoning paths by the Explainability Recall and Explainability Precision according to the top-3 matched courses.

Figure 2 shows the explainability of different path-reasoning methods on both datasets in terms of Explainability Recall and Explainability Precision. We only use ADAC as the comparison method since it is the state-of-the-art baseline for explainable recommendation. The explainability of our KRRL significantly outperforms ADAC on both MOOCCourse and MOOCCube datasets. Specifically, compared with the ADAC model, KRRL obtains significant improvements of 31.1% in terms of Explainability Recall@3 and 15.9% in terms of Explainability Precision@3 on the MOOCCourse dataset, while it improves 8.5% in terms of Explainability Recall@3 and 8.6% in terms of Explainability Precision@3 on the MOOCCube dataset. Combined with the results from Table 2, we can conclude that KRRL improves the recommendation accuracy while achieving better reasoning ability, compared with the competitive baselines. Besides, all the comparison methods achieve better explainability on MOOCCourse than that on MOOCCube. It indicates that richer associations on KGs contribute to the path reasoning, since the MOOCCourse dataset contains many more entities and relations than those from the MOOCCube dataset. In this case, the agent has more options to select the optimal action of path reasoning.

As observed from these results, KRRL is consistently superior to the other four variants on both datasets, in terms of Explainability Recall and Explainability Precision. For example, KRRL-D and KRRL-W are outperformed by the other two variants of KRRL in most cases, indicating that the self-supervised RL approach plays the most important role in path reasoning. Compared with KRRL and KRRL-P, KRRL-M has worse explainability, while it performs worse than KRRL-W in terms of Explainability Precision on the two datasets. This demonstrates that the multi-level representation method is also conducive to path reasoning over KGs. Additionally, the comparison of KRRL-P and KRRL shows that with the consideration of course prerequisite relations, KRRL can achieve better reasoning ability. This is because prerequisite relations among courses are helpful in capturing learners' knowledge levels. The experimental results further verify the effectiveness of these key components in our KRRL.

6.5 Case study (RQ5)

To intuitively show how the KRRL framework interprets the process of course recommendation, we conduct a case study by offering three real-world MOOCs examples of the path reasoning with KRRL. The results are illustrated in Fig. 3.

In the first example (Case 1) from the MOOCCourse dataset, a learner enrolled in a course *Circuit Analysis* belonged to the category of *Electronics*, which also

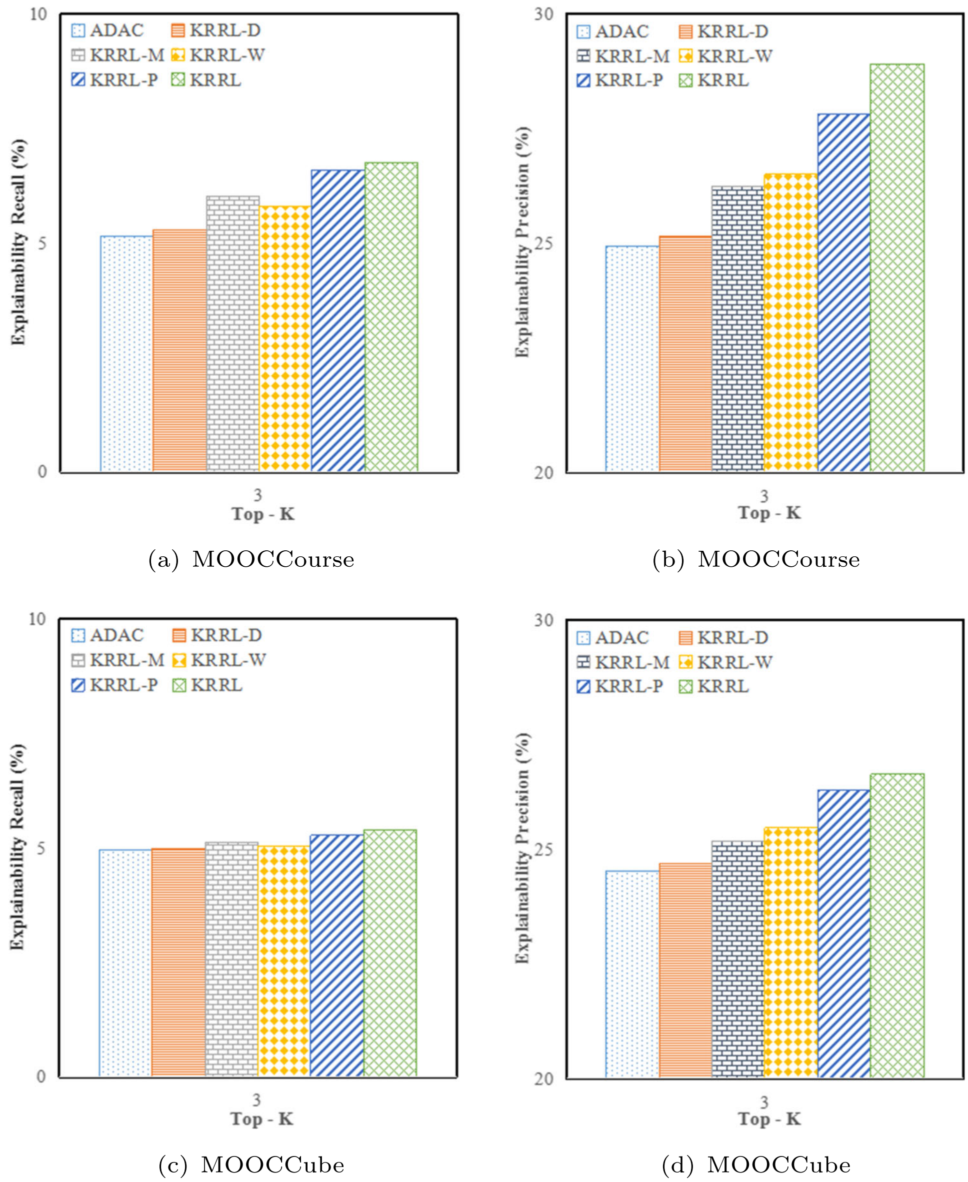

Fig. 2 Comparison of explainability on two MOOC datasets in terms of Explainability Recall and Explainability Precision (%)

includes the course *Digital Integrated Circuit Analysis and Design*. Hence, KRRL generated the reasoning path by using the demonstration in accord with a meta-path (*i.e.*, $user \xrightarrow{Enroll\_in} course \xrightarrow{Belong\_to} category \xleftarrow{Belong\_to} course$). Besides, it was also generated with another demonstration (*i.e.*, $user \xrightarrow{Enroll\_in} course \xrightarrow{Described\_by} course\ concept \xleftarrow{Described\_by} course$), since the two courses share some of the same concepts, such as “combinational circuits” and “resistance”, which can be represented as several knowledge points learned by the learner. This well explains why the learner may like the target course *Digital Integrated Circuit Analysis and Design*.

In the second example (Case 2) from the MOOCCourse dataset, both of learners A and B enrolled in the course *Web Development Technologies*, while learner B also enrolled in other courses. Thus, KRRL performs the path reasoning according to the meta-path $user \xrightarrow{Enroll\_in} course \xleftarrow{Enroll\_in} another\ user \xrightarrow{Enroll\_together} course$. It can be inferred that learner A may prefer the course *Java Programming* that learner B enrolled in, although learner B also enrolled in other courses, such as *Data Structures (I)* and *Career Exploration and Choice*. This is because the courses *Java Programming* and *Web Development Technologies* are closer in the semantic space, compared to the other courses that learner B enrolled in. Therefore, based on the weighted action path, the recommendation agent recommends the target course *Java Programming* to learner A.

The third example (Case 3) comes from the MOOCCube dataset. As shown in the bottom part of Fig. 3, a learner

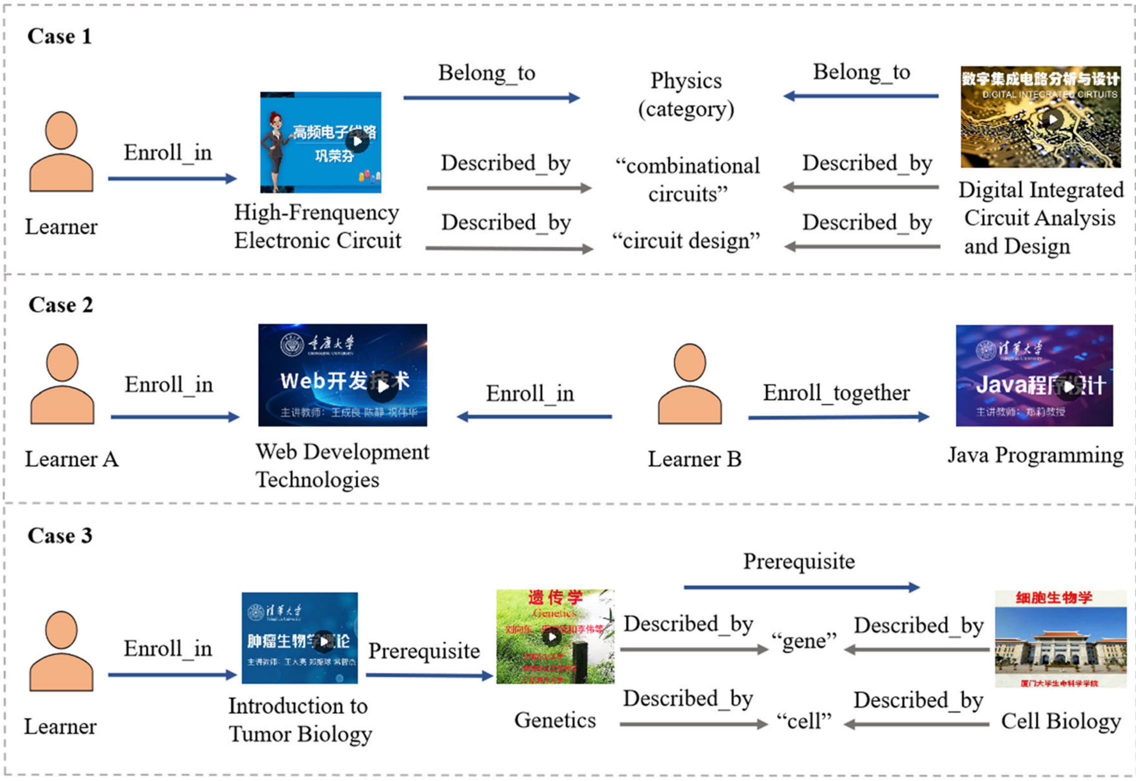

Fig. 3 Real cases of the path reasoning for course recommendation with our KRRL method

enrolled in a course *Introduction to Tumor Biology* that has a prerequisite course *Genetics*, which also has a prerequisite course *Cell Biology*. Thus, this reasoning path was generated with the demonstration that conforms to the meta-path $user \xrightarrow{Enroll\_in} course \xrightarrow{Prerequisite} course \xrightarrow{Prerequisite} course$. Meanwhile, the course *Genetics* contains two concepts "gene" and "cell", which are also described in the course *Cell Biology*. Hence, KRRL performs the path reasoning according to another meta-path $user \xrightarrow{Enroll\_in} course \xrightarrow{Described\_by} course\ concept \xleftarrow{Described\_by} course$. Such connectivity may reveal the potential factors in prerequisite relations among courses and can be effectively identified by the weighted operation to infer the learner's preference. Although *Genetics* has other prerequisite courses such as *Advanced Mathematics*, they do not share any course concepts. Therefore, KRRL recommended the target course *Cell Biology* to this learner.

Observed from these course examples of the path reasoning, it can be concluded that our KRRL can recommend suitable courses that not only are desirable (*i.e.*, personalization with learner's knowledge structure) but also satisfy constraints (*e.g.*, prerequisite requirements).

7 Conclusion and future work

7.1 Summary of the results and implications

In this paper, we have presented an explainable recommendation framework (KRRL), to address the issues related to recommendation accuracy and explainability under complex constraints in MOOCs.

We have discovered that the multi-level representation method not only improves the recommendation accuracy but also achieves good explainability. The main reason is that it introduces course concepts as auxiliary information to increase path connectivity. Thus, such connectivity enriches the perceptual information of semantic interactions in the KG and contributes to the path reasoning of course recommendation.

The experimental results demonstrate the effectiveness of the self-supervised RL approach. The effect of this approach is twofold: One is the weighted action path that infers learners' preferences, which contributes to the reasoning ability. The other is the IRL-based path discriminator, which finds rational demonstrations to enable accurate recommendations.

The empirical results also show that our proposed model actually interprets the process of course recommendation. Based on the conducted analysis of the case study, it indicates that KRRL is competent to conduct the knowledge-aware reasoning for qualified candidate paths, which match learners' knowledge structure and course prerequisite requirements.

The theoretical and practical implications of our research arise from the knowledge-aware reasoning method and system implementation for explainable recommendation based on the KG. Firstly, the multi-level representation learning method enhances the semantic representation and relation of KG. In fact, this method is general and it may also be applied to other KG-based recommendation models, since the existing KG-based recommendation models should learn more underlying user-item relations to improve the recommendation performance. Secondly, the self-supervised RL approach effectively guides the path reasoning over the KG. In particular, the weighted action paths can adjust the policy to infer learners' preferences. Moreover, the IRL algorithm employs expert demonstrations to enable accurate recommendations. We believe that the self-supervised RL approach can become a powerful tool to enhance the reasoning ability of KG-based recommendation models, because it conducts the path reasoning over the KG in a straightforward intelligent manner. The empirical findings from the analysis of the experimental results show that KRRL improves recommendation accuracy while achieving high-quality explainability.

7.2 Limitations and future work

There are still some limitations of the proposed framework that should be solved in future work. First of all, our KRRL framework only leverages the course concept and enrolled behaviors to learn the course and learners' representations. However, we do not take into account other auxiliary information, such as the course video and learners' comments. Besides, our method is limited to the explicit reasoning of MOOC recommendations, whereas the intrinsic mechanism of the proposed framework should be focused on.

For future work, we would like to investigate other learners' behaviors (*e.g.*, the duration of each course video watched by a learner) to mine the learner's potential preferences. Based on the information, it would be interesting to recommend various contents such as knowledge points and course videos to satisfy diverse requirements in course recommender systems [16]. Moreover, we plan to explore the interpretability of our KRRL framework, leveraging the causal inference mechanism [79] to increase its transparency.

Acknowledgements This research was supported by the National Natural Science Foundation of China under Grant Agreement No. 61977055.

Data availibility The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Huang T, Zhang D, Bi L (2020) Neural embedding collaborative filtering for recommender systems. Neural Comput Appl 32:17043–17057
2. Parameswaran A, Venetis P, Garcia-Molina H (2011) Recommendation systems with complex constraints: a course recommendation perspective. ACM Trans Inform Syst 29(4):1–33
3. Wu D, Lu J, Zhang G (2015) A fuzzy tree matching-based personalized e-learning recommender system. IEEE Trans Fuzzy Syst 23(6):2412–2426
4. Lin Y, Feng S, Lin F, Zeng W, Liu Y, Wu P (2021) Adaptive course recommendation in moocs. Knowl Based Syst 224:107085
5. Chen Z, Wang X, Xie X, Parsana M, Soni A, Ao X, Chen E (2020) Towards explainable conversational recommendation. In: Proceedings of the 29th International Joint Conference on Artifical Intelligence, pp 2994–3000
6. Zhao G, Fu H, Song R, Sakai T, Chen Z, Xie X, Qian X (2019) Personalized reason generation for explainable song recommendation. ACM Trans Intell Systd Technol 10(4):1–21
7. Lin Y, Lin F, Yang L, Zeng W, Liu Y, Wu P (2022) Context-aware reinforcement learning for course recommendation. Appl Soft Comput 125:109189
8. Ji S, Pan S, Cambria E, Marttinen P, Yu PS (2021) A survey on knowledge graphs: representation, acquisition, and applications. IEEE Trans Neural Netw Learn Syst 33(2):494–514
9. Li D, Madden A (2019) Cascade embedding model for knowledge graph inference and retrieval. Inform Process Manag 56(6):102093
10. Liu Y, Miyazaki J (2022) Knowledge-aware attentional neural network for review-based movie recommendation with explanations. Neural Comput Appl 32:1–19
11. Xian Y, Fu Z, Muthukrishnan S, de Melo G, Zhang Y (2019) Reinforcement knowledge graph reasoning for explainable recommendation. In: Proceedings of the 42nd International ACM SIGIR Conference, pp 285–294
12. Liu D, Lian J, Liu Z, Wang X, Sun G, Xie X (2021) Reinforced anchor knowledge graph generation for news recommendation reasoning. In: Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 1055–1065
13. Zhao K, Wang X, Zhang Y, Zhao L, Liu Z, Xing C, Xie X(2020) Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs. In: Proceedings of the 43rd International ACM SIGIR Conference, pp 239–248
14. Sutton RS, Barto AG (2018) Reinforcement Learning: An Introduction, 2nd edn. MIT, Massachusetts Ave, MA
15. Ruotsalo T, Weber S, Gajos K (2022) Active tag recommendation for interactive entity search: interaction effectiveness and retrieval performance. Inform Process Manag 59(2):102856
16. Jing X, Tang J (2017) Guess you like: course recommendation in moocs. In: Proceedings of the International Conference on Web Intelligence, pp 783–789

17. Wu D, Lu J, Zhang G (2015) A fuzzy tree matching-based personalized e-learning recommender system. IEEE Trans Fuzzy Syst 23(6):2412–2426
18. Xu J, Xing T, van der Schaar M (2016) Personalized course sequence recommendations. IEEE Trans Signal Process 64(20):5340–5352
19. Wan G, Pan S, Gong C, Zhou C, Haffari G (2021) Reasoning like human: hierarchical reinforcement learning for knowledge graph reasoning. In: Proceedings of the Twenty-ninth International Conference on International Joint Conferences on Artificial Intelligence, pp 1926–1932
20. Wang X, Wang D, Xu C, He X, Cao Y, Chua T.-S (2019) Explainable reasoning over knowledge graphs for recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 5329–5336
21. Ho J, Ermon S (2016) Generative adversarial imitation learning. In: Advances in Neural Information Processing Systems, pp 4565–4573
22. Fu Z, Xian Y, Gao R, Zhao J, Huang Q, Ge Y, Xu S, Geng S, Shah C, Zhang Y, de Melo G (2020) Fairness-aware explainable recommendation over knowledge graphs. In: Proceedings of the 43rd International ACM SIGIR Conference, pp 69–78
23. Suzuki T, Oyama S, Kurihara M (2019) Explainable recommendation using review text and a knowledge graph. In: 2019 IEEE International Conference on Big Data, pp 4638–4643
24. Tao S, Qiu R, Ping Y, Ma H (2021) Multi-modal knowledge-aware reinforcement learning network for explainable recommendation. Knowl Based Syst 227:107217
25. Geng S, Fu Z, Tan J, Ge Y, de Melo G, Zhang Y (2022) Path language modeling over knowledge graphs for explainable recommendation. In: Proceedings of the Web Conference, pp 946–955
26. Park S-J, Chae D-K, Bae H-K, Park S, Kim, S-W (2022) Reinforcement learning over sentiment-augmented knowledge graphs towards accurate and explainable recommendation. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, pp 784–793
27. Ma T, Huang L, Lu Q, Hu S (2023) Knowledge-aware reasoning with graph convolution network for explainable recommendation. ACM Trans Inform Syst 41(1):1–27
28. Li Q, Zhang Z, Zhuang F, Xu Y, Li C (2023) Topic-aware intention network for explainable recommendation with knowledge enhancement. ACM Trans Inform Syst 41(4):1–23
29. Yang Z, Dong S (2020) Hagerec: hierarchical attention graph convolutional network incorporating knowledge graph for explainable recommendation. Knowl Based Syst 204:106194
30. Wang P, Fan Y, Xia L, Zhao W.X, Niu S, Huang J (2020) Kerl: a knowledge-guided reinforcement learning model for sequential recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference, pp 209–218
31. Amador-Domínguez E, Serrano E, Manrique D (2023) Geni: a framework for the generation of explanations and insights of knowledge graph embedding predictions. Neurocomputing 521:199–212
32. Liu D, Lian J, Liu Z, Wang X, Sun G, Xie X (2021) Reinforced anchor knowledge graph generation for news recommendation reasoning. In: Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 1055–1065
33. Li H, Wang Y, Zhang S, Song Y, Qu H (2021) Kg4vis: a knowledge graph-based approach for visualization recommendation. IEEE Trans Visual Comput Graph 28(1):195–205
34. Huang J, Zhao W.X, Dou H, Wen J.-R, Chang E.Y (2018) Improving sequential recommendation with knowledge-enhanced memory networks. In: Proceedings of the 41nd International ACM SIGIR Conference, pp 505–514
35. Tang H, Zhao G, Bu X, Qian X (2021) Dynamic evolution of multi-graph based collaborative filtering for recommendation systems. Knowl-Based Syst 228:107251
36. Ai Q, Azizi V, Chen X, Zhang Y (2018) Learning heterogeneous knowledge base embeddings for explainable recommendation. Algorithms 11(9):137
37. Ma W, Zhang M, Cao Y, Jin W, Wang C, Liu Y, Ma S, Ren X (2019) Jointly learning explainable rules for recommendation with knowledge graph. In: The World Wide Web Conference, pp 1210–1221
38. Balloccu G, Boratto L, Fenu G, Marras M (2023) Reinforcement recommendation reasoning through knowledge graphs for explanation path quality. Knowl-Based Syst 260:110098
39. Huang Y, Zhao F, Gui X, Jin H (2021) Path-enhanced explainable recommendation with knowledge graphs. World Wide Web 24(5):1769–1789
40. Yang Z, Dong S, Hu J (2021) Gfe: general knowledge enhanced framework for explainable sequential recommendation. Knowl Based Syst 230:107375
41. Chen H, Li Y, Sun X, Xu G, Yin H (2021) Temporal meta-path guided explainable recommendation. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp 1056–1064
42. Wang X, Liu K, Wang D, Wu L, Fu Y, Xie X (2022) Multi-level recommendation reasoning over knowledge graphs with reinforcement learning. In: Proceedings of the Web Conference, pp 2098–2108
43. Tai C-Y, Huang L-Y, Huang C-K, Ku L-W (2021) User-centric path reasoning towards explainable recommendation. In: Proceedings of the 44th International ACM SIGIR Conference, pp 879–889
44. Khalid A, Lundqvist K, Yates A (2022) A literature review of implemented recommendation techniques used in massive open online courses. Expert Syst Appl 187:115926
45. Hoiles W, Schaar M.V.D (2016) Bounded off-policy evaluation with missing data for course recommendation and curriculum design. In: Proceedings of the 33rd International Conference on Machine Learning, pp 1596–1604
46. Wang C, Zhu H, Zhu C, Zhang X, Chen E, Xiong H (2020) Personalized employee training course recommendation with career development awareness. In: Proceedings of The Web Conference, pp 1648–1659
47. Li X, Li X, Tang J, Wang T, Zhang Y, Chen H (2020) Improving deep item-based collaborative filtering with bayesian personalized ranking for mooc course recommendation. In: Proceedings of International Conference on Knowledge Science, Engineering and Management, pp 247–258
48. Chang P, Lin C, Chen M (2016) A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems. Algorithms 9(3):47
49. Zhang J, Hao B, Chen B, Li C, Chen H, Sun J (2019) Hierarchical reinforcement learning for course recommendation in moocs. In: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, pp 435–442
50. Zhang H, Huang T, Lv Z, Liu S, Zhou Z (2018) Mcrs: a course recommendation system for moocs. Multimed Tools Appl 77(6):7051–7069
51. Ibrahim ME, Yang Y, Ndzi D (2017) Using ontology for personalised course recommendation applications. In: International Conference on Computational Science and Its Applications, pp 426–438
52. Dai Y, Asano Y, Yoshikawa M (2016) Course content analysis: an initiative step toward learning object recommendation systems for mooc learners. In: Proceedings of the 9th International Conference on Educational Data Mining, pp 347–352

53. Elbadrawy A, Karypis G (2016) Domain-aware grade prediction and top-n course recommendation. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp 183–190
54. Li X, Li X, Tang J, Wang T, Zhang Y, Chen H (2020) Improving deep item-based collaborative filtering with bayesian personalized ranking for mooc course recommendation. In: International Conference on Knowledge Science, Engineering and Management, pp 247–258
55. Tian X, Liu F (2021) Capacity tracing-enhanced course recommendation in moocs. IEEE Trans Learn Technol 14(3):313–321
56. Chen Q, Yu X, Liu N, Yuan X, Wang Z(2020) Personalized course recommendation based on eye-tracking technology and deep learning. In: IEEE 7th International Conference on Data Science and Advanced Analytics, pp 692–698
57. Zhang J, Le T, Liao Y, Lee D (2019) Pathfinder: Graph-based itemset embedding for learning course recommendation and beyond. In: International Conference on Data Mining Workshops, pp 1122–1125
58. Chen H, Yin C, Fan X, Qiao L, Rong W, Zhang X (2021) Learning path recommendation for mooc platforms based on a knowledge graph. In: International Conference on Knowledge Science, Engineering and Management, pp 600–611
59. Wang X, Ma W, Guo L, Jiang H, Liu F, Xu C (2022) Hgnn: hyperedge-based graph neural network for mooc course recommendation. Inform Proces Manag 59(3):102938
60. Chen W, Niu Z, Zhao X, Li Y (2014) A hybrid recommendation algorithm adapted in e-learning environments. World Wide Web 17(2):271–284
61. Esteban A, Zafra A, Romero C (2020) Helping university students to choose elective courses by using a hybrid multi-criteria recommendation system with genetic optimization. Knowle Based Syst 194:105385
62. Zhu Y, Lu H, Qiu P, Shi K, Chambua J, Niu Z (2020) Heterogeneous teaching evaluation network based offline course recommendation with graph learning and tensor factorization. Neurocomputing 415:84–95
63. Konda VR, Tsitsiklis JN (2003) On actor-critic algorithms. Siam J Control Optim 42(4):1143–1166
64. Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O (2013) Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, pp 2787–2795
65. Pan L, Li C, Li J, Tang J (2017) Prerequisite relation learning for concepts in moocs. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp 1447–1456
66. Wan S, Niu Z (2016) A learner oriented learning recommendation approach based on mixed concept mapping and immune algorithm. Knowl-Based Syst 103:28–40
67. Dijkstra EW (1959) A note on two problems in connexion with graphs. Numer Math 1:269–271
68. Sutton RS (1988) Learning to predict by the methods of temporal differences. Mach Learn 3:9–44
69. Sutton RS, McAllester DA, Singh SP, Mansour Y (2000) Policy gradient methods for reinforcement learning with function approximation. In: Advances in Neural Information Processing Systems, pp 1057–1063
70. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2016) Continuous control with deep reinforcement learning. In: Proceedings of the International Conference on Learning Representations, pp 1–14
71. Bellman R (1966) Dynamic programming. Science 3731(153):34–37
72. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp 452–461
73. McAuley J, Leskovec J (2013) Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of the 7th ACM Conference on Recommender Systems, pp 165–172
74. He X, Deng K, Wang X, Li Y, Zhang Y, Wang M (2020) Lightgcn: simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 639–648
75. Wang J, Xie H, Wang FL, Lee L-K, Au OTS (2021) Top-n personalized recommendation with graph neural networks in moocs. Comput Educ: Artif Intell 2:100010
76. Zhang Y, Chen X (2020) Explainable recommendation: a survey and new perspectives. Found Trends Inform Retrieval 14(1):1–101
77. Zhang W, Lin Y, Liu Y, You H, Wu P, Lin F, Zhou X (2022) Self-supervised reinforcement learning with dual-reward for knowledge-aware recommendation. Appl Soft Comput 131:109745
78. Abdollahi B, Nasraoui O (2017) Using explainability for constrained matrix factorization. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp 79–83
79. Yao L, Chu Z, Li S, Li Y, Gao J, Zhang A (2021) A survey on causal inference. ACM Trans Knowl Discov Data 15(5):1–46