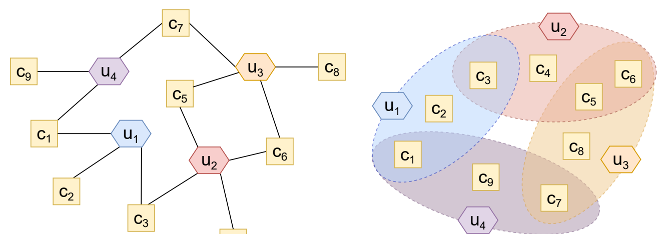
Hierarchical Reinforcement Learning on Multi-Channel Hypergraph Neural Network for Course Recommendation

Lu Jiang[1,4], **Yanan Xiao**[1,4], **Xinxin Zhao**[1,4], **Yuanbo Xu**[2,5], **Shuli Hu**[1,4], **Pengyang Wang**[3,6*] and **Minghao Yin**[1,4*]

[1]School of Computer Science and Information Technology, Northeast Normal University, China
[2]College of Computer Science and Technology, Jilin University, China
[3]Department of Computer and Information Science, University of Macau, China
[4]Key Laboratory of Applied Statistics of MOE, Northeast Normal University, China
[5]Mobile Intelligent Computing (MIC) Lab, Jilin University, China
[6]The State Key Laboratory of Internet of Things for Smart City, University of Macau, China
{jiangl761,xiaoyn117, zhaoxx767, husl903, ymh}@nenu.edu.cn, yuanbox@jlu.edu.cn, pywang@um.edu.mo

Abstract

With the widespread popularity of massive open online courses, personalized course recommendation has become increasingly important due to enhancing users' learning efficiency. While achieving promising performances, current works suffering from the vary across the users and other MOOC entities. To address this problem, we propose **H**ierarchical reinforcement learning with a multi-channel **H**ypergraphs neural network for **Co**urse **R**ecommendation (called **HHCoR**). Specifically, we first construct an online course hypergraph as the environment to capture the complex relationships and historical information by considering all entities. Then, we design a multi-channel propagation mechanism to aggregate embeddings in the online course hypergraph and extract user interest through an attention layer. Besides, we employ two-level decision-making: the low-level focuses on the rating courses, while the high-level integrates these considerations to finalize the decision. Finally, we conducted extensive experiments on two real-world datasets and the quantitative results have demonstrated the effectiveness of the proposed method.



(a) Heterogeneous graph (b) Hypergraph

Figure 1: The differences between a heterogeneous graph (a) and a hypergraph (b). Figure (a) shows an edge connecting two nodes, while figure (b) shows an example of users' hypergraph with 9 courses and 4 hyperedges.

1 Introduction

The prosperity of massive open online courses (MOOCs) is due to the rapid development of online education. The overwhelming and spotty learning materials in MOOC platforms undermine users' efficiency. Against this background, accurately modeling user preference for learning materials offers valuable insights with course recommender system [Zhang *et al.*, 2019]. The selection of the next course by users is influenced by the interplay between network interactions, which echo user needs and vary. Therefore, in this paper, we propose to develop an effective recommender system with hypergraph learning for course recommendation in MOOCs.

Prior literature in an online course recommendation method can be categorized into three aspects: (1) Collaborative filtering (CF) method [Yang and Cai, 2022] relies on user-item interaction data to predict course preferences; (2) Sequence-based method [Shao *et al.*, 2021; Hou *et al.*, 2018] uses the sequence of courses to recommend future learning paths; (3) Graph-based method [Wang *et al.*, 2021; Xu *et al.*, 2022] uses a complex network structure to model the relationship between users and courses. There are two main challenges: (1) the interactions among users are very complex and the relationships can be high-order; and (2) traditional recommendations cannot model real-time online study behavior in a continuously updated manner. Below we formally introduce each challenge and how we address them in our proposed framework.

First, graph neural network (GNN)-based [Wang *et al.*, 2021] models have shown promising performance in course recommendation, due to the powerful capability in modeling relationships. A limitation of these GNN-based recommendation methods is that exploit the pairwise relations and ig-

*Corresponding author.

nore the high-order relations among the entities. Although the long dependencies of relations are considered high-order, which can be captured by using k-hop node neighbors, these only permit a maximum of two entities per relationship, as shown in Figure 1(a). These heterogeneous graph structures are unable to formulate complex high-order user relations beyond pairwise relations. Hypergraph [Fan *et al.*, 2021] can capture high-order relationships by allowing edges to connect more than two nodes. As shown in Figure 1(b), it is natural to think that two users who are studying the same course have a stronger relationship, we employ hypergraph to make it connect more than two nodes, to model complex high-order relations among users. We define the MOOC hypergraph to organize the multiple to multiple relationships. We utilize hyperedges to mine high-order semantic information between various types entity to form multiple channels. And incorporates an attention mechanism in the information transmission process to ensure semantic integrity during cross perspectives information propagation. By aggregating multiple embeddings learned through multiple channels, we can obtain comprehensive user representations that are considered to contain multiple types of high-order relations.

Second, it is natural and promising to exploit reinforcement learning, a real-time learning paradigm optimized with long-term reward, to develop a course recommender system for MOOCs. To achieve this goal, we reformulate the course recommendation problem in MOOC as a hierarchical reinforcement learning task. HHCoR is built following the two-layer decision-making process: (1) the low level focuses on the rating courses, and (2) the high level integrates these considerations to finalize the decision. To facilitate our framework with a proper environment, we propose a MOOC hypergraph to organize the multi-channel semantics of study records. The hyperedge embeddings from this MOOC hypergraph serve as the state to support the decision-making process in our method. In summary, we formulate the online course recommendation problem as **H**ierarchical reinforcement learning with multi-channel **H**ypergraphs neural network for **Co**urse **R**ecommendation (called **HHCoR**).

The main contributions are as follows:

- We reformulate the problem of personalized course recommendation as a task based on hierarchical reinforcement learning.
- We construct a MOOC hypergraph, which effectively handles the heterogeneous nature of courses and utilizes an attention mechanism to capture user preferences from multi-channel semantics.
- We design a policy optimization framework based on hierarchical reinforcement learning and introduce reward function guidance mechanism to optimize the two-level agent's policy.
- We validate our method on two real datasets and the results demonstrate the excellent performance of our method on the task of course recommendation.

2 Definitions and Problem Formulation

2.1 MOOC Hypergraph

In order to capture the complex relationships between the participation of multiple entities on the MOOC platform, we propose to construct a hypergraph to represent historical records, called MOOC Hypergraph. Formally, MOOC Hypergraph $\mathcal{G}$ is defined as $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ and $\mathbf{E}$ represents the vertex set and hyperedge set respectively. Each hyperedge $e \in \mathbf{E}$ connects two or more vertices.

Vertices. MOOC hypergraphs aim to organize MOOC elements while preserving multi-aspect semantics. Specifically, we categorize MOOC elements into three semantic channels, including (1) the course channel, denoted as $\mathbf{c}$; (2) the concept channel, denoted as $\mathbf{k}$; (3) the video channel, denoted as $\mathbf{o}$. In this work, we consider three types of vertices corresponding to three semantic channels. Then, the vertex set can be denoted as $\mathbf{V} = \mathbf{c} \cup \mathbf{k} \cup \mathbf{o}$.

Hyperedge. We define four types of hyperedges: (1) Course hyperedge, which connects to all course nodes that the user has been enrolled in; (2) Concept hyperedge, which connects all learned concept nodes; (3) Video hyperedge, which connects the video nodes that the user has watched; (4) Feature hyperedge, connecting user, concept, and video nodes to each other. We learn user perspectives from multiple sources, and user perspectives consist of four types of hyperedge embeddings. We utilize the Parallel Aggregated Ordered Hypergraph [Valdivia *et al.*, 2021] (PAOH) model to construct our proposed MOOC hypergraph and hyperedges.

2.2 Problem Formulation

In this work, we formulate course recommendation as a Markov Decision Process [Feinberg and Shwartz, 2012] (MDP). Users decide which course to enroll in next based on a history that reflects their personal preferences under a particular MOOC platform. The main components of the MDP are defined as (1) **States** S. Each state $\mathbf{s} \in S$ represents a specific user context derived from the MOOC platform history, which is organized into a MOOC hypergraph. (2) **Actions** A. Each action $a \in A$ corresponds to a potential next enrollment course. (3) **Transition Probabilities** Γ. $\Gamma(\mathbf{s}'|\mathbf{s}, a)$ denotes the probability of transitioning from state $\mathbf{s}$ to state $\mathbf{s}'$ when action a is taken. This probability can be estimated from the user's platform history and reflects how often the user transitions from one learning environment to another after selecting a particular course. (4) **Rewards** R. $R(\mathbf{s}, a, \mathbf{s}')$ denotes the reward received after transitioning from state $\mathbf{s}$ to state $\mathbf{s}'$ due to action a. The reward can be designed to reflect user satisfaction or any other metric of interest. We will introduce the reward design later. (5) **Environment** E. The environment consists of all participants of study events. It responds to the user's action by providing a new state and a reward. The environment's dynamics are governed by the transition probabilities Γ and the reward function R. (6) **Policy** π. A policy π defines how users take action. Specifically, $\pi(\mathbf{s})$ gives the probability distribution over actions in state $\mathbf{s}$. The goal of the MDP is to find an optimal policy π^* that maximizes the expected cumulative reward over time.

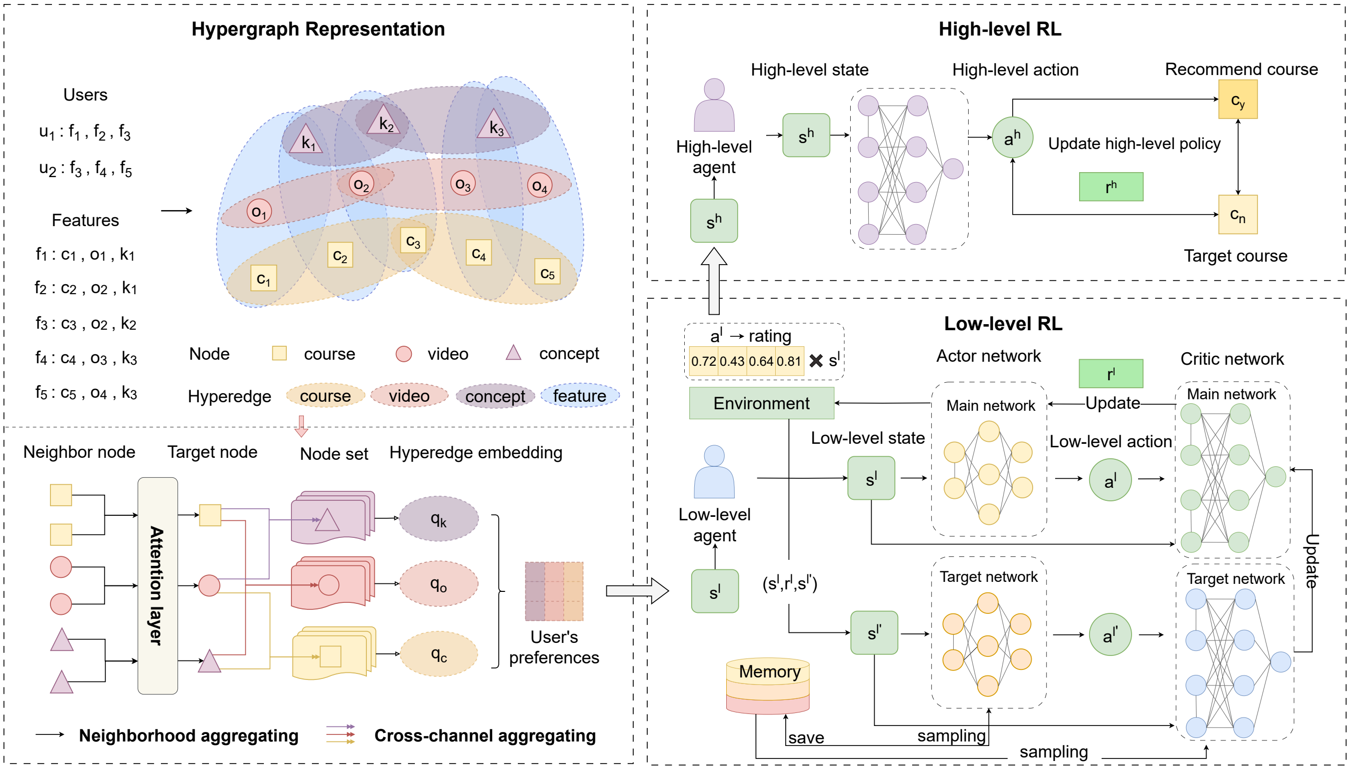


Figure 2: Framework Overview.

In view of the course being studied the form of the MDP is recommended, our goal is to develop a hierarchical reinforcement learning framework to find the optimal policy π^* that guides the user's decision to register for the next course.

3 Method

In this section, we introduce the core architecture of our method HHCoR, including hypergraph representation learning, low-level policy, and high-level policy.

3.1 Framework Overview

The proposed HHCoR is illustrated in Figure 2. First, we learn the state of the environment by constructing a MOOC hypergraph, we propose a multi-channel aggregating mechanism to propagate various information among nodes in three channels. Then, we utilize the attention layer to extract the user preferences based on different hyperedges. After that, the low-level agents take the environment state as input, and the low-level agents model the multidimensional preference representation by analyzing the importance of each historical course to the target course. Finally, the high-level agents formulate a course recommendation policy by receiving learning insights from the low-level agents. The two-layer agents reinforce each other through iterative updates.

3.2 Hypergraph Representation Learning

Vertex Embedding. We denote the raw features of vertex $v_i \in \mathcal{V}$ as $\mathbf{x}_i \in \mathbb{R}^d$, and $\mathcal{N}_i$ represents vertex v_i's neighbors that are within the hyperedges. We employ the attention mechanism to capture the interrelationship between vertices and the respective neighbors in the same channel. Specifically, for the vertex v_i and its neighbor v_j ($j \in \mathcal{N}_i$), the attention coefficient α_{ij} can be represented as

$$\alpha_{ij} = \frac{\exp(\mathbf{v}_i \mathbf{v}_j)}{\sum_{v_j \in \{\mathcal{N}_i, i\}} \exp(\mathbf{v}_i \mathbf{v}_j)}. \tag{1}$$

Then, the embedding $\mathbf{h}_i$ of the node v_i can be represented by aggregate the neighbors' define as

$$\mathbf{h}_i = \sum_i \alpha_{ij} \mathbf{v}_i. \tag{2}$$

Hyperedge Embedding. In our study, we defined four types of hyperedges, including courses, videos, concepts, and features. Among them, course, video, and concept hyperedges are homogeneous (connecting vertices within the same semantic channel) and feature hyperedges are heterogeneous (connecting vertices across all semantic channels). For the homogeneous hyperedge $e_i \in \mathbf{E}$, we denote the hyperedge embedding by the set of all node embeddings within the hyperedge. The hyperedge embedding $\mathbf{q}_i$ can be represented as

$$\mathbf{q}_i = \sigma\Big(\sum_{j \in |e_i|} \mathbf{h}_j\Big), \tag{3}$$

where $|e_i|$ denotes all the linked nodes in e_i.

The feature hyperedges serve as a bridge to link the semantics from different perspectives through the hypergraph topology. We then update the hyperedge embedding $\mathbf{q}_i$ by aggregating information from hyperedges on other perspectives that are interlinked by the same feature hyperedge:

$$\mathbf{q}_i = \sigma \left(\sum_{k \in \Phi(e_i)} \mathbf{W}_{\Psi(e_k)} \mathbf{q}_k \right), \tag{4}$$

Where σ is the sigmoid function [Elfwing *et al.*, 2018], $\Phi(e_i)$ denotes the query function that retrieves hyperedges from alternate perspectives that are interconnected by the same feature hyperedge as the given hyperedge, $\Psi(\cdot)$ is the function to return the type of the given hyperedge, and $\mathbf{W}_{\Psi(e_k)}$ denotes the aggregation weights for the given the type $\Psi(e_k)$.

3.3 Low-level Policy

In the initial phase of the HHCoR system, the low-level agent is responsible for meticulously rating historical courses, and this rating process is a key foundation for understanding and recognizing user decision-making patterns. Subsequent sections will detail the core components and operational mechanisms that make up low-level decision-making.

State. We use hyperedge embeddings as a representation of states. Specifically, for the low-level agent, states aim to capture the preferences and interactions of multiple aspects of the MOOC platform. Therefore, we connect relevant hyperedge embeddings to represent the state. Formally, let $\mathbf{s}^l$ denote the low-level agent state define as

$$\begin{aligned} &\mathbf{s}^l = \text{CONCATENATE}(\mathbf{q}_c, \mathbf{q}_k, \mathbf{q}_o) \\ &\text{c} = \Theta_{\mathbf{c}}(u) \ \& \ \text{k} = \Theta_{\mathbf{k}}(u) \ \& \ \text{o} = \Theta_{\mathbf{o}}(u), \end{aligned} \tag{5}$$

where $\Theta_{\mathbf{c}}(u)$, $\Theta_{\mathbf{k}}(u)$ and $\Theta_{\mathbf{o}}(u)$ denote the indexes of associated course hyperedge, concept hyperedge, and video hyperedge for the user u, respectively.

Low-Level Agent with DDPG. In the HHCoR framework, the low-level agent comprises two parts: the 'critic', which assesses historical courses by computing the value function $Q(s, a|\theta^Q)$ for each action, and the 'actor', which refines strategies based on these evaluations. This process involves scoring predictions to reflect the effectiveness of course actions, with the output—a weight between 0 and 1—indicating each course's significance for user representation. The value function is defined as

$$Q(s^l, a^l|\theta^Q) \approx Q^*(s^l, a^l), \tag{6}$$

Where $Q^*(s^l, a^l)$ represents the optimal action-value function. The critic network is trained by minimizing a defined loss function defined as

$$L(\theta^Q) = \mathbb{E}_{s^l, a^l, r^l, s^{l'}}[(Q(s^l, a^l|\theta^Q) - y)^2], \tag{7}$$

Where $y = r^l + \gamma Q(s^{l'}, a^{l'}|\theta^Q)$ is the target value, γ denotes discount factor emphasizing the importance of future rewards and $s^{l'}$ and $a^{l'}$ represent the next state and action respectively.

In the actor component, another neural network is used to approximate the policy with parameters θ^μ defined as

$$\mu(s^l|\theta^\mu) \approx \pi^{l^*}(s^l), \tag{8}$$

where $\pi^*(s^l)$ is the optimal policy.

The actor-network is trained by applying the policy gradient [Kakade, 2001] defined as

$$\nabla_{\theta^\mu} J \approx \mathbb{E}_{s^l}[\nabla_{\theta^\mu} \mu(s^l|\theta^\mu) \nabla_{a^l} Q(s^l, a^l|\theta^Q)]. \tag{9}$$

Then, to enhance the exploration capabilities of our model, we introduce the controllable stochasticity [Lapan, 2018] to promote exploration. Specifically, we use the Ornstein-Uhlenbeck [Lillicrap *et al.*, 2016] process to generate temporally correlated noise.

Low-level Reward Function. The reward function for low-level agents is intended to guide learning. The reward r^l is computed as the change in correlation between the target predicted value and the real enrolled course before and after the action a^l, defined as

$$r^l = Q(s^{l'}, a^{l'}|\theta^Q) - Q(s^l, a^l|\theta^Q), \tag{10}$$

where the agent's action a^l outputs a probability ranging from 0 to 1, indicating the current course's relevance to the user's historical preferences.

If a low-level agent's action a^l improves the relevance of a target course's prediction, it earns a positive reward; otherwise, a negative reward is given for reduced relevance. This incentivizes the agent to adjust the importance weights of historical courses, enhancing predictive accuracy. Continuous interaction with the environment and corresponding rewards enable the agent to develop effective course rating strategies, thus aiding the decision-making of high-level agents.

3.4 High-level Policy

The high-level decision-making process employs a specialized agent to amalgamate insights garnered from lower-level agents, effectively merging these insights with platform factors within the MOOC hypergraph framework. This integration facilitates the formulation of a comprehensive course recommendation decision. This section delineates the principal components of the high-level agent and provides an overview of its operational workflow.

State. In order to encapsulate the low-level agent's understanding of the user's preference, the state of the high-level agent is defined by the updated low-level agent state defined as

$$\begin{aligned} &\mathbf{s}^h = \text{CONCATENATE}(\mathbf{q}'_{\mathbf{c}}, \mathbf{q}'_{\mathbf{k}}, \mathbf{q}'_{\mathbf{o}}) \\ &\text{c} = \Theta_{\mathbf{c}}(u) \ \& \ \text{k} = \Theta_{\mathbf{k}}(u) \ \& \ \text{o} = \Theta_{\mathbf{o}}(u), \end{aligned} \tag{11}$$

where $\mathbf{q}'_{\mathbf{c}}$, $\mathbf{q}'_{\mathbf{k}}$ and $\mathbf{q}'_{\mathbf{o}}$ denote the relevant course hyperedge, concept hyperedge, and video hyperedge embeddings of user u after the low-level agent update.

High-Level Agent with REINFORCE. The high-level agent implements the REINFORCE algorithm [Williams, 1992], utilizing feedback from the low-level agent and environmental data for prediction guidance. This agent adopts a stochastic policy $\pi^h(s^h, a^h|\theta^{\pi^h})$, with s^h and a^h denoting the state and action, respectively, aimed at forecasting the user's next likely course selection. The policy parameters θ^{π^h} are refined through gradient ascent define as

$$\begin{aligned} \nabla_{\theta^{\pi^h}} J \approx \mathbb{E}_{s^h, a^h}[&\nabla_{\theta^{\pi^h}} \log \pi^h(s^h, a^h|\theta^{\pi^h}) \\ &\cdot (Q^h(s^h, a^h) - b(s^h))], \end{aligned} \tag{12}$$

Datasets	MOOCCube						MOOCCourse					
Metrics	HR			NDCG			HR			NDCG		
Baselines	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
FISM	0.1254	0.2001	0.3187	0.0800	0.1039	0.1336	0.2584	0.3925	0.5779	0.1758	0.2189	0.2655
MLP	0.1939	0.3006	0.4498	0.1233	0.1576	0.1951	0.4874	0.6306	0.7790	0.3532	0.3994	0.4370
NAIS	0.1194	0.1956	0.3123	0.0758	0.1004	0.1296	0.2642	0.4042	0.5875	0.1753	0.2202	0.2664
HRL	0.2580	0.4027	0.6116	0.1609	0.2075	0.2600	0.6543	0.8061	0.8796	0.4717	0.5216	0.5403
SR-GNN	0.0881	0.1360	0.2386	0.0636	0.0788	0.1041	0.2441	0.3024	0.3759	0.1792	0.2179	0.2364
LightGCN	0.1488	0.2024	0.3411	0.0822	0.0933	0.2422	0.2704	0.4412	0.6645	0.1994	0.2645	0.2933
COTREC	0.0823	0.1336	0.1960	0.0440	0.0605	0.0762	0.2046	0.2623	0.3392	0.1017	0.1201	0.1395
DHCN	0.1272	0.1856	0.2508	0.0927	0.1115	0.1279	0.1973	0.2416	0.3139	0.1463	0.1604	0.1786
CoHHN	0.2776	0.4316	0.6355	0.2230	0.2370	0.2460	0.5514	0.6837	0.7991	0.4236	0.4931	0.5525
HHCoR	**0.3477**	**0.5140**	**0.7420**	**0.2241**	**0.2816**	**0.3135**	**0.6985**	**0.8351**	**0.8932**	**0.5041**	**0.5635**	**0.5830**

Table 1: Overall Performance Comparison.

Where $Q^h(s^h, a^h)$ is the action-value function as estimated by the high-level agent and $b(s^h)$ is a baseline function for variance reduction. We adopt the mean of the action-value function as this baseline function. The high-level agent processes the output of the low-level agent along with the environmental information to make its decisions.

Exploring Deterministic and Stochastic Policies. We explore two policies for the high-level agent: a deterministic policy and a stochastic policy.

- **Stochastic Policy**: By employing the REINFORCE algorithm, Advanced Agents adopt a random strategy to introduce a certain degree of randomness in course selection. This approach facilitates deeper exploration of the course catalog to uncover hidden preferences or interests of users.
- **Deterministic Policy**: Conversely, we implement a deterministic policy for the high-level agent where it consistently recommends the same courses in response to specific user profiles or behaviors. This approach ensures stability and efficiency, focusing on optimizing user satisfaction with highly relevant courses, although it may limit the variety of courses explored.

High-level Reward Function. We developed a reward function r^h for the high-level agent, aimed at enhancing its decision-making accuracy. This function comprises three components: (1) Concept similarity r_k between the target and predicted courses, ; (2) Video content similarity r_o between the target and predicted courses; and (3) The probability of recommending the target course r_p. The overall reward is a combination of these elements defined as

$$r^h = w_k \cdot r_k + w_o \cdot r_o + w_p \cdot r_p, \tag{13}$$

where w_k, w_o, and w_p denote the weights for balancing the influence of r_k, r_o, r_p, respectively.

This weighting allows for fine-tuning of the recommendation process, ensuring that each aspect of the user's preferences is appropriately considered, leading to highly personalized and effective course recommendations.

4 Experiment

In our study, we carried out a comprehensive series of experiments on two real-world MOOC datasets to address five key research questions:

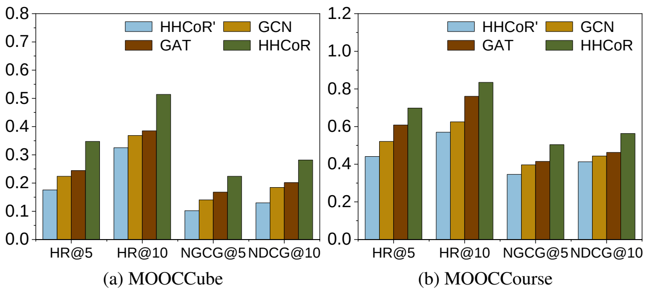
- **Q1:** How is the performance of our proposed HHCoR in the course recommendation task?
- **Q2:** How does the MOOC hypergraph affect HHCoR recommendation performance?
- **Q3:** How does the MOOC hyperedge affect HHCoR recommendation performance?
- **Q4:** How do different components of the agent contribute to decision-making in our model?
- **Q5:** How do different reward designs impact course recommendation performance?

4.1 Experiment Settings

Datasets. We evaluate the model performance using two datasets: the MOOCCube [Yu *et al.*, 2020] and the MOOCCourse [Zhang *et al.*, 2019; Lin *et al.*, 2022]. The samples in the training and test sets consist of a sequence of historical courses with the target course. For training, the last course in the sequence is the target course and the rest are history courses. Each positive sample corresponds to the construction of four negative samples that replace the target course. For testing, the course in the test set was used as the target and paired with 99 random negative samples.

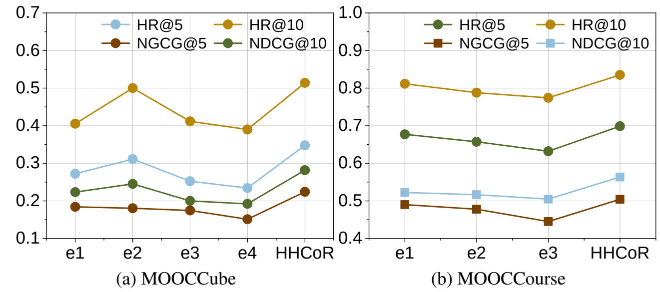
Baselines. We compare our proposed method with the following baseline algorithms, including (1) **FISM** [Kabbur *et al.*, 2013]; (2) **MLP** [He *et al.*, 2017]; (3) **NAIS** [He *et al.*, 2018]; (4) **HRL** [Zhang *et al.*, 2019]; (5) **SR-GNN** [Wu *et al.*, 2019]; (6) **LightGCN** [He *et al.*, 2020]; (7) **DHCN** [Xia *et al.*, 2021b]; (8) **COTREC** [Xia *et al.*, 2021a] and (9) **CoHHN** [Zhang *et al.*, 2022].

Evaluation Metrics. We evaluate the course recommendation accuracy in terms of the widely used metrics, including hit ratio (HR@N) and normalized discounted cumulative gain (NDCG@N). Evaluation was performed with N = 5, 10, 20.



(a) MOOCCube (b) MOOCCourse

Figure 3: An ablation study on hypergraph.



(a) MOOCCube (b) MOOCCourse

Figure 4: An ablation study on hyperedge types.

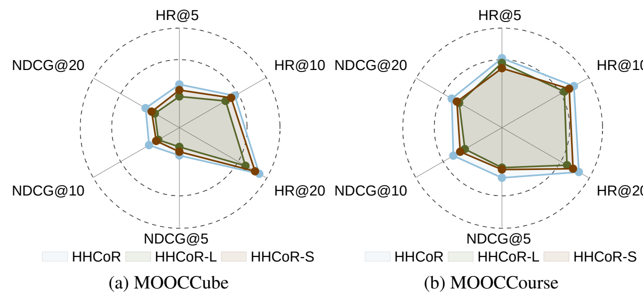
Hyperparameter Settings. For the hypergraph representation, the dimensionality of the node embeddings was set to 64 and we utilized 8 attention heads in the attention mechanism. The DDPG agent and the REINFORCE agent were optimized with a discount factor (γ) set to 0.99. Both the agents employed Adam optimizers, with the learning rates set to 0.001.

4.2 Overall Performance (Q1)

In this section, we compare the overall performance of all models on real datasets. Overall, as Table 1 indicates, our model outperforms other baselines in HR and NDCG metrics. Compared with MLP representing node attributes, item-based collaborative filtering methods (FISM, NAIS), reinforcement learning-based methods (HRL), and graph neural network-based methods (SR-GNN, LightGCN, COTREC, DHCN, CoHHN), Our proposed method incorporates course-related auxiliary information, which is more comprehensive and performs better in capturing users' interests. Compared with item-based collaborative filtering methods and reinforcement learning-based methods, our proposed framework also considers heterogeneous hypergraph embeddings and high-order semantic relations between heterogeneous information. Compared to graph neural network-based methods, our proposed method analyzes the degree to which each historical course of a user represents that user's interests. In conclusion, the results validate that our model is beneficial for course recommendation, which can help to better infer users' interests and improve recommendation accuracy.

4.3 The Study of MOOC Hypergraph (Q2)

We conducted an experiment to verify the necessity of the hypergraph structure. In this experiment, we designed a variant of HHCoR, called (HHCoR', which directly takes the user's sequence as input without using the hypergraph structure. Beyond that, we replaced the hypergraph representation with other well-known graph representations such as Graph Convolutional Network (GCN) and Graph Attention Network (GAT). As shown in Figure 3, HHCoR exhibits a significant performance advantage. The superiority of HHCoR over its variants underscores the unique ability of hypergraph architectures to model complex relationships and higher-order interactions among data points, which standard graph models like GCN and GAT might miss.



(a) MOOCCube (b) MOOCCourse

Figure 5: An ablation study of the low-level agent.

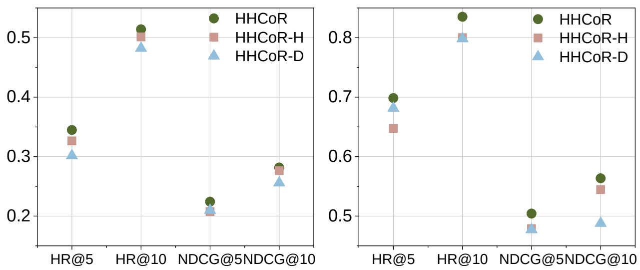
4.4 The Study of MOOC Hyperedges (Q3)

In the MOOCCube dataset, we conducted experiments to assess hyperedge types' impact, including the removal of concept (e_1), video (e_2), feature (e_3) hyperedges individually, and removing all except the course hyperedge (e_4). For the MOOCCourse dataset, experiments involved removing field (e_1) and feature (e_2) hyperedges, and a combined removal of field and feature (e_3). As shown in Figure 4, the performance of different hyperedge combinations varies, highlighting their importance in capturing the multi-semantics of users on MOOC platforms. HHCoR achieves optimal performance when it incorporates all types of hyperedges.

4.5 The Study of Agent Architecture (Q4)

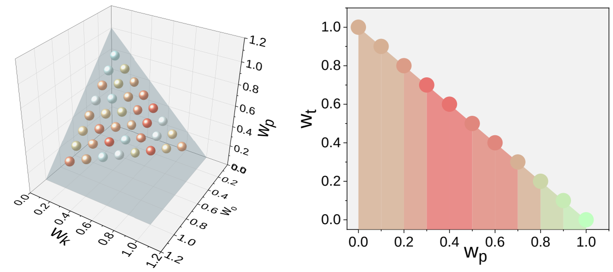
The design of Low-level Agent. The results from HHCoR-L, where the low-level agent is omitted, indicate a marked reduction in the system's capacity for precise user preference analysis, highlighting the agent's integral role in processing course-related data. In the case of HHCoR-S, restricting the agent's exploration scope leads to a diminished ability to generate innovative recommendations, crucial for adaptive learning. As Figure 5, these outcomes not only validate the essential role of the low-level agent in the HHCoR framework but also underscore its contribution to the sophistication and reliability of the course recommendation process.

The design of High-level Agent. Our study examined the significance of the high-level agent in our hierarchical reinforcement learning model through two experiments: HHCoR-H, which omits the high-level agent's explicit predictive function, and HHCoR-D, employing a deterministic policy for the high-level agent. These tests, results of which are depicted in Figure 6, aimed to assess the influence of the high-level agent's predictive capacity and policy randomness



(a) MOOCCube (b) MOOCCourse

Figure 6: An ablation study of the high-level agent.



(a) MOOCCube (b) MOOCCourse

Figure 7: The analysis of reward of the high-level agent.

on model performance. The findings confirm that the high-level agent's explicit prediction, stochastic policy, and collaborative reward mechanism are integral to the overall effectiveness and robustness of our model.

4.6 The Study of Reward Design (Q5)

We consider combinations of weight settings for high-level agents and different reward functions to test the performance of HHCoR. The low-level reward is automatically learned and cannot be manually adjusted. Therefore, we only analyze the reward settings of the high-level agent. In our analysis, MOOCCube considers three components: w_k, w_o, and w_p, while MOOCCourse involves two components: w_t and w_p. We mapped the performance of various combinations (where $w_k + w_o + w_p = 1$ for MOOCCube, and $w_t + w_p = 1$ for MOOCCourse) onto 3D and 2D spaces, respectively. As shown in Figure 7, the better the performance, the darker the color.

5 Related Work

5.1 Personalized Course Recommendation

Personalized course recommendation has advanced from traditional content-based and collaborative filtering, which struggles with scalability and capturing dynamic user preferences, to more sophisticated machine learning techniques. These include matrix factorization, factorization machines, and deep learning methods like autoencoders and RNNs, which better handle complex user-course interactions. Studies like [Hou *et al.*, 2018; Xu *et al.*, 2024; Xu *et al.*, 2022; Yang and Jiang, 2019] have made notable contributions, utilizing course clusters and combined user-course networks, respectively. Despite improvements, these methods still face challenges in adapting to the evolving and varied preferences of online learning users.

5.2 Graph-based Methods in Course Recommendation

Graph-based methods like GCN have been increasingly applied in personalized course recommendation to address its challenges. Studies like [Wang *et al.*, 2021; Zhu *et al.*, 2023a; Wang *et al.*, 2022] effectively utilize these techniques for capturing intricate user-course relationships, with the latter viewing user embeddings as hyperedges in a learning hypergraph. Such methods excel in identifying complex, high-order relationships, a feat traditional methods often miss. However, they typically assume a homogeneous graph structure, which doesn't align with the heterogeneous nature of MOOCs. To address this, [Fan *et al.*, 2021; Xia *et al.*, 2022] have explored the use of heterogeneous hypergraphs and hypergraph transformer networks, respectively, offering a more fitting solution for modeling the diverse and complex relationships prevalent in MOOC platform.

5.3 Reinforcement Learning in Course Recommendation

Reinforcement learning (RL) in course recommendation treats it as a sequential decision-making problem, adept at handling dynamic user behavior for optimized long-term suggestions. [Gong *et al.*, 2022; Zhu *et al.*, 2020; Zhu *et al.*, 2023b] approached MOOC recommendations using RL, employing meta-paths on HIN and a heterogeneous graph attention network. Similarly, [Jiang *et al.*, 2023] used a MOOC knowledge graph to guide interpretable recommendation paths. Traditional RL, however, struggles with large, complex action spaces typical in course recommendations, necessitating the use of Hierarchical Reinforcement Learning (HRL). [Xie *et al.*, 2021; Zhang *et al.*, 2024; Zhao *et al.*, 2020] tackled this by dividing the recommendation process into multiple tasks, with agents operating at different abstraction levels, thereby effectively managing personalized and multi-objective recommendations.

6 Conclusion

In this paper, we study the problem of personalized course recommendation with a MOOC hypergraph and propose a hierarchical reinforcement learning framework for multi-channel hypergraph neural networks for online course recommendation. Specifically, we first formulate the MOOC personalized recommendation problem as a task based on hierarchical reinforcement learning. Secondly, we construct a MOOC hypergraph and propose to use the attention mechanism to extract the multi-channel semantics of MOOC entity relationships in different channels and capture user preferences. Third, we design a policy optimization framework based on hierarchical reinforcement learning and introduce reward function guidance mechanism to optimize the two-level agent's policy. Finally, we conduct extensive experiments on two real-world MOOC datasets to verify the effectiveness of our proposed method.

Acknowledgments

This work is supported by NSFC(under Grant No. 62106040, 61976050) , Jilin Province Science and Technology Department Project (under Grant No. YDZJ202201ZYTS415, 20240602005RC), Jilin Education Department Project under Grant No.JJKH20231319KJ, Jilin Science and Technology Association under Grant No. QT202320, and the Fundamental Research Funds for the Central Universities No.2412022ZD016, JLU. This work is surported by the Science and Technology Development Fund (FDCT), Macau SAR (file no. 0123/2023/RIA2, 001/2024/SKL), the Start-up Research Grant of University of Macau (File no. SRG2021-00017-IOTSC).

References

[Elfwing *et al.*, 2018] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

[Fan *et al.*, 2021] Haoyi Fan, Fengbin Zhang, Yuxuan Wei, Zuoyong Li, Changqing Zou, Yue Gao, and Qionghai Dai. Heterogeneous hypergraph variational autoencoder for link prediction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(8):4125–4138, 2021.

[Feinberg and Shwartz, 2012] Eugene A Feinberg and Adam Shwartz. *Handbook of Markov decision processes: methods and applications*, volume 40. Springer Science Business Media, 2012.

[Gong *et al.*, 2022] Jibing Gong, Yao Wan, Ye Liu, Xuewen Li, Yi Zhao, Cheng Wang, Yuting Lin, Xiaohan Fang, Wenzheng Feng, Jingyi Zhang, et al. Reinforced moocs concept recommendation in heterogeneous information networks. *ACM Trans. Web*, 2022.

[He *et al.*, 2017] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *WWW*, pages 173–182, 2017.

[He *et al.*, 2018] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. Nais: Neural attentive item similarity model for recommendation. *IEEE Trans. Knowl. Data Eng.*, 30(12):2354–2366, 2018.

[He *et al.*, 2020] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*, pages 639–648, 2020.

[Hou *et al.*, 2018] Yifan Hou, Pan Zhou, Jie Xu, and Dapeng Oliver Wu. Course recommendation of mooc with big data support: A contextual online learning approach. In *INFOCOM WKSHPS*, pages 106–111. IEEE, 2018.

[Jiang *et al.*, 2023] Lu Jiang, Kunpeng Liu, Yibin Wang, Dongjie Wang, Pengyang Wang, Yanjie Fu, and Minghao Yin. Reinforced explainable knowledge concept recommendation in moocs. *ACM Trans. Intell. Syst. Technol.*, 14(3):1–20, 2023.

[Kabbur *et al.*, 2013] Santosh Kabbur, Xia Ning, and George Karypis. Fism: Factored item similarity models for top-n recommender systems. In *KDD*, pages 659–667, 2013.

[Kakade, 2001] Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

[Lapan, 2018] Maxim Lapan. *Deep Reinforcement Learning Hands-On: Apply modern RL methods, with deep Q-networks, value iteration, policy gradients, TRPO, AlphaGo Zero and more*. Packt Publishing Ltd, 2018.

[Lillicrap *et al.*, 2016] T Lillicrap, J Hunt, Alexander Pritzel, N Hess, Tom Erez, D Silver, Y Tassa, and D Wiestra. Continuous control with deep reinforcement learning. In *ICRL*, 2016.

[Lin *et al.*, 2022] Yuanguo Lin, Fan Lin, Lvqing Yang, Wenhua Zeng, Yong Liu, and Pengcheng Wu. Context-aware reinforcement learning for course recommendation. *Applied Soft Computing*, 125:109189, 2022.

[Shao *et al.*, 2021] Erzhuo Shao, Shiyuan Guo, and Zachary A Pardos. Degree planning with plan-bert: Multi-semester recommendation using future courses of interest. In *AAAI*, volume 35, pages 14920–14929, 2021.

[Valdivia *et al.*, 2021] Paola Valdivia, Paolo Buono, Catherine Plaisant, Nicole Dufournaud, and Jean-Daniel Fekete. Analyzing dynamic hypergraphs with parallel aggregated ordered hypergraph visualization. *IEEE Trans. Vis. Comput. Graph.*, 27(1):1–13, 2021.

[Wang *et al.*, 2021] Jingjing Wang, Haoran Xie, Fu Lee Wang, Lap-Kei Lee, and Oliver Tat Sheung Au. Top-n personalized recommendation with graph neural networks in moocs. *Computers and Education: Artificial Intelligence*, 2:100010, 2021.

[Wang *et al.*, 2022] Xinhua Wang, Wenyun Ma, Lei Guo, Haoran Jiang, Fangai Liu, and Changdi Xu. Hgnn: Hyperedge-based graph neural network for mooc course recommendation. *Inf. Process. Manag.*, 59(3):102938, 2022.

[Williams, 1992] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[Wu *et al.*, 2019] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *AAAI*, volume 33, pages 346–353, 2019.

[Xia *et al.*, 2021a] Xin Xia, Hongzhi Yin, Junliang Yu, Yingxia Shao, and Lizhen Cui. Self-supervised graph co-training for session-based recommendation. In *CIKM '21*, page 2180–2190, New York, NY, USA, 2021. ACM.

[Xia *et al.*, 2021b] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. Self-supervised hypergraph convolutional networks for session-based recommendation. In *AAAI*, volume 35, pages 4503–4511, 2021.

[Xia *et al.*, 2022] Lianghao Xia, Chao Huang, and Chuxu Zhang. Self-supervised hypergraph transformer for recommender systems. In *KDD*, pages 2100–2109, 2022.

[Xie *et al.*, 2021] Ruobing Xie, Shaoliang Zhang, Rui Wang, Feng Xia, and Leyu Lin. Hierarchical reinforcement learning for integrated recommendation. In *AAAI*, volume 35, pages 4521–4528, 2021.

[Xu *et al.*, 2022] Yuanbo Xu, En Wang, Yongjian Yang, and Yi Chang. A unified collaborative representation learning for neural-network based recommender systems. *IEEE Trans. Knowl. Data Eng.*, 34(11):5126–5139, 2022.

[Xu *et al.*, 2024] Yuanbo Xu, En Wang, Yongjian Yang, and Hui Xiong. GS-RS: A generative approach for alleviating cold start and filter bubbles in recommender systems. *IEEE Trans. Knowl. Data Eng.*, 36(2):668–681, 2024.

[Yang and Cai, 2022] Shuang Yang and Xuesong Cai. Bilateral knowledge graph enhanced online course recommendation. *Information Systems*, 107:102000, 2022.

[Yang and Jiang, 2019] Xixi Yang and Wenjun Jiang. Dynamic online course recommendation based on course network and user network. In *iSCI*, pages 180–196. Springer, 2019.

[Yu *et al.*, 2020] Jifan Yu, Gan Luo, Tong Xiao, Qingyang Zhong, Yuquan Wang, Wenzheng Feng, Junyi Luo, Chenyu Wang, Lei Hou, Juanzi Li, et al. Mooccube: a large-scale data repository for nlp applications in moocs. In *ACL*, pages 3135–3142, 2020.

[Zhang *et al.*, 2019] Jing Zhang, Bowen Hao, Bo Chen, Cuiping Li, Hong Chen, and Jimeng Sun. Hierarchical reinforcement learning for course recommendation in moocs. In *AAAI*, volume 33, pages 435–442, 2019.

[Zhang *et al.*, 2022] Xiaokun Zhang, Bo Xu, Liang Yang, Chenliang Li, Fenglong Ma, Haifeng Liu, and Hongfei Lin. Price does matter! modeling price and interest preferences in session-based recommendation. In *SIGIR*, pages 1684–1693, 2022.

[Zhang *et al.*, 2024] Zhaofan Zhang, Yanan Xiao, Lu Jiang, Dingqi Yang, Minghao Yin, and Pengyang Wang. Spatial-temporal interplay in human mobility: A hierarchical reinforcement learning approach with hypergraph representation. pages 9396–9404. AAAI Press, 2024.

[Zhao *et al.*, 2020] Dongyang Zhao, Liang Zhang, Bo Zhang, Lizhou Zheng, Yongjun Bao, and Weipeng Yan. Mahrl: Multi-goals abstraction based deep hierarchical reinforcement learning for recommendations. In *SIGIR*, pages 871–880, 2020.

[Zhu *et al.*, 2020] Yifan Zhu, Hao Lu, Ping Qiu, Kaize Shi, James Chambua, and Zhendong Niu. Heterogeneous teaching evaluation network based offline course recommendation with graph learning and tensor factorization. *Neurocomputing*, 415:84–95, 2020.

[Zhu *et al.*, 2023a] Yifan Zhu, Fangpeng Cong, Dan Zhang, Wenwen Gong, Qika Lin, Wenzheng Feng, Yuxiao Dong, and Jie Tang. WinGNN: dynamic graph neural networks with random gradient aggregation window. In *The 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023*. ACM, 2023.

[Zhu *et al.*, 2023b] Yifan Zhu, Qika Lin, Hao Lu, Kaize Shi, Donglei Liu, James Chambua, Shanshan Wan, and Zhendong Niu. Recommending learning objects through attentive heterogeneous graph convolution and operation-aware neural network. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):4178–4189, 2023.