International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18 (2026) pp. 1-20

Article

A model construction of personalized teaching of English writing based on reinforcement learning

Shali Jin[1], SANITAH BTE MOHD[1,*], Norliza Mohamad[1] and Abdulmumini Inda[1]

[1] Universiti Teknologi Malaysia, Johor Bahru, Johor, 81310, Malaysia
* Correspondence author: jsl900912@163.com

Abstract: By constructing a deep knowledge tracking model based on reinforcement learning theory, students can choose more suitable courses or exercises when selecting learning resources. In view of this, the article proposes a temporal convolutional knowledge tracking model (TCKT-FI) based on reinforcement learning, which obtains students' knowledge mastery state through temporal convolutional network, solves the long sequence dependency problem by using causal convolution and inflationary convolution, and solves the gradient vanishing and exploding problems by using residual network to deal with the deep network structure. Following that, a recommendation algorithm framework based on Deep Reinforcement Learning (DRR) is proposed, and better recommendation strategies (Actor) and value functions (Critic) are explored. Finally, the model's English writing knowledge tracking effect is tested and its performance is verified through experiments. The knowledge tracking effect test experiment shows that at moments 1.1~5, 5.8-9.5, 10.5-15, when students did not learn the knowledge point, the TCKT-FI model shows a significant downward trend, indicating that students have forgotten the knowledge point, but the comparison model fails to simulate the forgetting process of students well. This confirms the accuracy of the model and the efficiency of its calculation. It has practical value in the personalized teaching of English writing.

Keywords: reinforcement learning; knowledge tracking; temporal convolutional network; English writing

1. Introduction

English writing teaching is an important part of cultivating students' comprehensive language use ability and a key indicator of their English proficiency [1]. With the in-depth promotion of the new curriculum reform, the goal of writing teaching has shifted from pure language skills training to the comprehensive improvement of core literacy, with special emphasis on the cultivation of thinking quality, cultural awareness and learning ability [2]. However, the traditional writing teaching mode faces many challenges, such as the long feedback cycle of teacher's correction, single teaching resources, obsolete teaching methods, and insufficient evaluation system, which seriously restricts the effective enhancement of students' writing ability, and this contradictory relationship urgently needs to be reconciled through the innovation of teaching paradigm [3]. The rapid development of artificial intelligence (AI) provides a new opportunity for teaching reform. Its powerful data analysis and intelligent interaction capabilities are expected to break through the predicament of English writing teaching and open up a new path of personalized learning, which has become the focus of current research.

At present, personalized teaching has become a popular teaching tool and occupies an important position in China's new curriculum reform [4]. At the same time, it is also one of the directions of education and teaching reform in various countries. As an important auxiliary means of personalized teaching, artificial intelligence can facilitate the implementation of personalized English writing teaching. Therefore, many scholars at home and abroad have carried out relevant research. For instance,

Mourtzis et al. within the framework of "Teaching Factory" proposed and verified a hybrid teaching model. This model expanded the dimensions of teaching, not only including technology, innovative teaching methods and learning environment, but also introducing two key elements of continuous support and cybersecurity, emphasizing multi-level collaboration and the construction of a healthy digital learning environment, providing a reference practical path for personalized teaching [5]. Bhutoria's research found that AI can effectively adapt to students' individual learning needs, habits, and abilities, guiding them into optimized learning paths, while enabling the enhancement and personalization of educational content and providing early warnings of potential learning difficulties, thus reshaping the role of the teacher and optimizing the teaching and learning environment [6]. Hu verified the significant advantages of the data-driven personalized teaching mode in enhancing student engagement and learning effectiveness by constructing a fine-grained learner portrait and implementing personalized resource recommendations, demonstrating clear effectiveness compared to traditional teaching methods, providing empirical support for AI-enabled personalized education [7]. Li constructed a multi-objective classroom teaching decision-making optimization model for students' individualized learning needs, which takes maximizing classroom teaching quality, maximizing the attention to individualized learning needs and maximizing the degree of student dissatisfaction as the optimization objectives, and gives the corresponding solution method, which provides quantitative support for optimizing classroom teaching decision-making to satisfy students' individualized needs [8]. Wu et al. analyzed the two core dimensions of student modeling and personalized recommendation, explored the interaction mechanism of the two in enhancing the effect of personalized learning, and systematically sorted out the application effectiveness of personalized learning in real educational scenarios, providing a multidimensional perspective for the integration of the theory and practice of personalized learning in the context of intelligent education [9].

The rapid development of AI technology has triggered strong interest among scholars at home and abroad in exploring its application in personalized English language teaching. Zhou and Li proposed an AI-based Learning-Teaching Model (AI-L-TM) for recommending learning paths focusing on analyzing the learning performance and acquiring new knowledge.The model utilizes Internet of Things (IoT) devices, data mining methods, and classroom data collection with the aim of optimizing the English learning experience for students in higher education [10]. Wang proposed a deep collaborative learning resource recommendation model based on the attention mechanism, aiming to promote the development of intelligent English education; the study pointed out that the Internet of Things (IoT) senses and collects data through multiple sensors, and builds network connections using multiple communication technologies, which provides a basis for effective collection of educational data, and is an important support for the personalized English education to achieve teaching, management, and resource recommendation [11]. Yuan proposed a personalized college English adaptive learning method based on artificial intelligence, which collects student data through intelligent sensing devices and combines big data analysis and machine learning technology to achieve real-time personalized customization of the English learning experience [12]. Zhan focused on the construction and application of a personalized university English teaching model based on big data under the empowerment of the intelligent platform, aiming to enhance teaching quality and effectiveness through data-driven teaching innovation; by analyzing the differences in students' learning styles, interests, knowledge foundation, and abilities, a complete framework from data collection to teaching implementation was constructed, and then personalized teaching plans were formulated and transformed into specific teaching practices [13].

Regarding the application of personalized teaching model in English writing teaching, Dasam et al. proposed a personalized writing feedback system based on the T5 model, aiming at solving the shortcomings of traditional feedback methods in terms of timeliness, precision and individual adaptability, which utilizes the text-to-text framework of T5 to provide comprehensive feedback covering grammatical corrections, stylistic optimization, and overall writing enhancement, and is capable of generating real-time and contextually relevant instruction based on learners' needs to generate real-time and contextually relevant instruction [14]. Begum et al. proposed a deep learning system based on the BERT-LSTM model for providing real-time feedback for English writing in online learning environments to solve the problem of inconsistent quality in traditional writing feedback due to differences in teachers' time and ability; the system combines BERT's ability to analyze linguistic details with LSTM's modeling advantages of sequential information and contextual context to generate personalized feedback based on the The system combines BERT's ability to analyze linguistic details and LSTM's advantage of modeling sequence information and contextual context to generate personalized feedback based on learners' writing behaviors and styles [15]. Hwang et al. developed the Smart Roam Lingo application, which provides contextualized samples based on recognition technology and personalized revision suggestions based on students' original texts through two

functions, AI Sample Sentences (AI-SS) and AI Writing Feedback (AI-WF), respectively, in order to cope with the problem of insufficient vocabulary resources and the lack of personalized feedback in English-as-a-foreign-language writing [16]. Palacios et al. explored teachers' usage and perceptions of the use of ChatGPT in writing instruction and found that ChatGPT has certain advantages as a personalized learning aid, but there are also concerns about the ethical risks it poses and the potential threat to critical thinking, resulting in four instructional contexts: the Ideal Zone, the Tension Balance Zone, the Opportunity Zone, and the Critical Zone [17].

In various studies by scholars at home and abroad, different algorithms have also been proposed from different perspectives for the personalized teaching model of learners, and in-depth analyses, validations and discussions have been carried out, which provide us with certain inspiration for the construction of the personalized teaching model of English writing [18]. The development of reinforcement learning (RL) algorithms brings data-driven accurate solutions for the construction of personalized teaching models, and its intelligent decision-making mechanism can dynamically adjust the training strategy and improve the training effect. Reinforcement learning is a machine learning method for solving sequential decision-making problems, which learns the optimal strategy by allowing an intelligent body to interact with the environment to obtain feedback [19-21]. Regarding the application of RL algorithms in personalized learning, Tang et al. constructed the problem of automatic recommendation of learning sequences in personalized learning as a Markov decision-making process and proposed a solution method based on reinforcement learning, which utilizes a data-driven recommender system that seeks a balance between utilizing existing knowledge and exploring potentially better learning paths in order to achieve personalized recommendation of learning materials and optimization of learning effectiveness [22]. Shawky and Badawi proposed an adaptive learning system construction method based on reinforcement learning, which is not only capable of recommending appropriate learning materials, but also dynamically adapts to the continuous changes in the learner's state and his/her acceptance of the technology, and is applicable to a variety of contexts such as individual learning and collaborative learning [23]. Sharif and Uckelmann proposed the KNIGHT framework which integrates multimodal data, innovatively applies deep reinforcement learning to educational analytics, validates the effectiveness of the framework through case studies, and demonstrates its potential to drive change in personalized education [24]. Amin et al. proposed an intelligent e-learning framework based on reinforcement learning with Markov decision processes, and the model demonstrated significant performance enhancement under different parameter optimization conditions and outperformed traditional methods in long sequence learning contexts, validating its effectiveness in providing personalized and adaptive learning paths [25]. Su proposed a personalized recommendation algorithm for educational content based on deep reinforcement learning, aiming to solve the problem of inaccurate recommendation caused by the difficulty of extracting feature vectors of educational content during the recommendation process, and optimize the final output by personalized algorithm [26].

The article firstly provides an overview of the relevant basics of reinforcement learning and a detailed introduction to reinforcement learning algorithms, including their basic concepts, AC framework and DDPG algorithm. Then, a temporal convolutional network is used as the infrastructure for modeling, and a temporal convolutional knowledge tracking model (TCKT-FI) integrating forgetting factor and IRT is proposed. The model introduces time convolutional network TCN instead of recurrent neural network to solve the problem of long problem sequence dependency, while the time convolutional network applies residual network to effectively solve the problem of gradient vanishing and gradient explosion that may occur in the deep network structure of the model. To further explore more accurate modeling methods for user state representation in reinforcement learning-based recommendation models. The article proposes a deep reinforcement learning recommendation framework, DRR, based on Actor-Critic architecture.Within the DRR framework, a user state representation learning module is proposed and four different network structures are designed to model the user state representation through user-item interaction information. In the experiments, the article conducts a large number of experiments on four learning datasets. It verifies the effectiveness of this article's model for tracking students' English writing status. Meanwhile, the experiments on two different diverse assessment indicators based on the learning datasets verify the advantages of the method proposed in this paper over the existing methods.

2. Method

2.1. Enhanced learning theory

Reinforcement learning (RL) learns decision-making styles in different environmental states during

interaction with the environment for the purpose of maximizing long-term rewards. Although traditional reinforcement learning is limited to a limited action space and state space, it is difficult to solve complex problems in the real world. Deep Reinforcement Learning (DRL) utilizes the powerful feature extraction capability of deep learning tools to perceive the state or encode the action space in complex environments, enabling reinforcement learning techniques to solve complex problems in real-world scenarios, and is considered to be a pathway towards general artificial intelligence.

2.1.1. Enhanced Learning Basic Concepts

Reinforcement learning theory draws on human behavioral psychology to describe and solve the problem of maximizing long-term rewards through learning strategies that intelligences use in their interactions with the environment. The basic principle is that if an intelligent's behavior according to a certain strategy leads to high rewards from the environment, the intelligent will be more inclined to utilize this strategy in similar environments in the future. The round of interaction between an intelligent body and the environment is divided into three steps: first, the intelligent body senses the state of the environment. Then, the intelligent body makes an action based on the current state. Next, the state of the environment may change due to the execution of the action, and a real-valued reward is given. The intelligent body learns a strategy to maximize the cumulative reward as it interacts continuously with the environment.

The process can be described using a Markov Decision Process (MDP), which is shown in Figure 1.

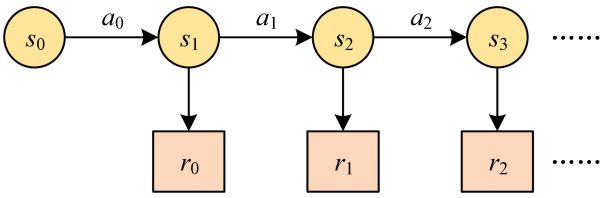

Figure 1. Markov decision process

SMDP can be represented as a quintuple $\{S, A, P_a, R_a, \gamma\}$. Where S represents the state space. A represents the action space. P_a represents the probability of transferring to state s' after taking action a in state s, as shown in equation (1). R_a represents the immediate reward received after taking action a in state s and transferring to state s', denoted as $R_a(s,s')$. The γ is a real number in the $(0,1)$ interval representing the discount factor in calculating the long-term reward, as shown in equation (2). The interval restriction on γ is mainly considered in view of the uncertainty of future rewards and to enable the summation to converge:

$$P_a(s,s') = P\left(s_{t+1} = s' \middle| s_t = s, a_t = a\right) \tag{1}$$

$$R_{long} = \sum_{t=1}^{\infty} \gamma^t R_{a_t}\left(s_t, s_{t+1}\right) \tag{2}$$

The core of the MDP problem is to solve the optimal policy. There are two types of strategies: deterministic and uncertain strategies.

The former of these can be expressed as $a = \pi(s)$, where the action obtained at each state is unique. The latter is denoted as $a = \pi\left(a \middle| s\right)$, where actions are made at each state according to a certain probability distribution.

If the state in which the action is performed to move to the next step is deterministic, the optimized long-term reward can be computed directly according to equation (1). However, most problems do not fulfill this condition, and in order to measure the merit of the strategy, the concept of value function is defined to measure the long-term reward expected to be obtained. The value function is divided into state value function $V_\pi(s)$ and action value function $Q_\pi(s,a)$. According to the definition of value function, the value function has the following recursive formula:

$$V_\pi(s) = \sum_a \pi\left(a \middle| s\right) \sum_{s'} P_a(s,s')\left(R_a(s,s') + \gamma V_\pi(s')\right) \tag{3}$$

$$Q_\pi(s,a) = \sum_{s'} P_a(s,s')\left(R_a(s,s') + \gamma V_\pi(s')\right) \tag{4}$$

The above formulation is known as the Bellman equation and is the core of the Markov decision process. There are three methods for finding the optimal policy: value iteration, policy iteration and policy search. Among them, value iteration and policy iteration iteratively calculate the value function and then find the optimal policy, while policy search directly optimizes the objective function of reinforcement learning:

$$J(\theta) = \sum_{\tau} P(\tau;\theta)R(\tau) \tag{5}$$

where θ denotes the parameters of the non-deterministic strategy $\pi_\theta(a|s)$ or the deterministic strategy $\pi_\theta(a)$, τ denotes a set of state-action sequences $(s_0, a_0, s_1, \ldots, a_{N-1}, s_N)$, $P(\tau|\theta)$ denotes the probability of the occurrence of the trajectory, and $R(\tau)$ denotes the total rewards obtained in the trajectory. The most common method used to optimize this objective function is gradient descent, at which point the strategy search is referred to as the strategy gradient. The parameter update formula for the strategy gradient is as follows:

$$\theta = \theta - \eta \nabla_\theta J(\theta) \tag{6}$$

where $\nabla_\theta J(\theta)$ is calculated as follows:

$$\begin{aligned} \nabla_\theta J(\theta) &= \nabla \sum_{\tau} P(\tau;\theta)R(\tau) \\ &= \sum_{\tau} \nabla P(\tau;\theta)R(\tau) = \sum_{\tau} \frac{P(\tau;\theta)}{P(\tau;\theta)} \nabla P(\tau;\theta)R(\tau) \\ &= \sum_{\tau} P(\tau;\theta) \nabla_\theta \log(P(\tau;\theta))R(\tau) \\ &= E_{P(\tau;\theta)}\left(\nabla_\theta \log(P(\tau;\theta))R(\tau)\right) \end{aligned} \tag{7}$$

In practice, this can be estimated by taking an empirical average:

$$\nabla_\theta J(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log(P(\tau_i;\theta))R(\tau_i) \tag{8}$$

where τ_i is the i th empirical trajectory and there are m empirical trajectories in total.

2.1.2. Actor-Critic Framework

As can be seen from the formula for the strategy gradient, the direction of the gradient of the objective function contains two factors: $\nabla_\theta \log(P(\tau;\theta))$ and $R(\tau)$. Where the former value represents the gradient of $\log(P(\tau;\theta))$ with respect to θ. The latter is the reward received by the intelligence and is proportional to the update magnitude. Thus the intuitive meaning of the strategy gradient is to update the strategy parameters in the direction that maximizes the reward. The strategy gradient can be expressed as:

$$g = E\left[\sum_{t=0}^{\infty} \psi_t \nabla_\theta \left(\log \pi_\theta\left(a_t|s_t\right)\right)\right] \tag{9}$$

2.1.3. Deep deterministic strategy gradient

The DDPG algorithm is an algorithm based on the AC framework, where the Actor and Critic parts use neural networks to fit $\pi(s)$ and $Q_\pi(s_t, a_t)$, respectively. The algorithm draws on the experience playback mechanism of DQN and the setting of the target network.

The experience playback mechanism mainly relies on the experience pool structure implementation, in which the experience information (s_t, a_t, r_t, s_{t+1}) of one interaction is stored. Whenever an intelligent body and the environment do an interaction, a set of experience information is generated. After the intelligent body and the environment do N times of interaction, n sets of experience information are randomly taken out from the experience pool to train the neural network. This design disrupts the correlation of the training data and makes the training process more efficient.

The design of the deep reinforcement learning network uses two sets of networks with the same structure but different parameters, the primary network and the target network. Where the primary network is responsible for interacting with the environment, the target network is responsible for memorizing the parameter updating process and providing the next Q value prediction. This design

can keep the target Q -value is constant in a certain time, which improves the stability of the algorithm.

The DDPG algorithm flow is shown in Figure 2.

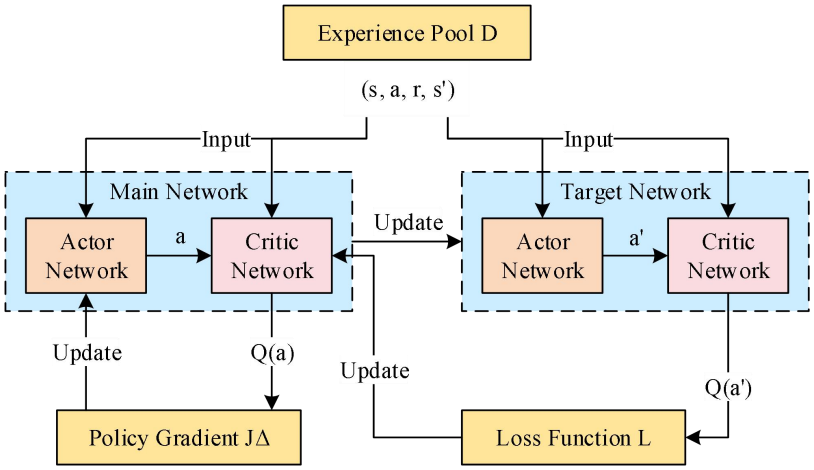

Figure 2. DDPG algorithm process

After a set of samples are randomly batch drawn from the experience pool, the Actor network first takes s_t as input and outputs the strategy vector a_t to represent the action, and then s_t and a_t are jointly inputted into the Critic network to obtain $Q(s_t, a_t)$, and the parameters of the Actor network are updated according to the strategy gradient ∇J, in which the empirical gradient is found as in equation (10):

$$\nabla J = \frac{1}{N}\sum_i \nabla_a Q(s,a)\Big|_{s=s_i, a=\pi(s_i)} \nabla_{\theta_\pi} \pi(s)\Big|_{s=s_i} \tag{10}$$

s_{t+1} is used as an input to the target network, and $Q(s_{t+1}, a_{t+1})$ is obtained in the same way, updating the Critic network in the main network according to the loss function in Eq. (11):

$$L = \frac{1}{N}\sum_i \left(r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right)^2 \tag{11}$$

Whenever the primary network updates N step, the target network updates its own parameters using its parameter part:

$$\theta_\pi \leftarrow \tau\theta_\pi + (1-\tau)\theta_{\pi,target} \tag{12}$$

$$\theta_Q \leftarrow \tau\theta_Q + (1-\tau)\theta_{Q,target} \tag{13}$$

where θ_π and θ_Q are the parameters of the Actor and Critic networks in the primary network, $\theta_{\pi,target}$ and $\theta_{Q,target}$ are the parameters of the Actor and Critic networks in the target network, respectively. parameters, and τ is a real number between the intervals $(0,1)$.

2.2. Time-convolutional knowledge tracking model construction incorporating forgetting factor and IRT

The structure of TCKT-FI model is shown in Fig. 3. First, the interaction data between students and exercises under 2D convolution were extracted using TCN to generate the knowledge state module. Then, the forgetting factors affecting the forgetting behavior were abstracted using the psychology of forgetting and integrated into the forgetting factors module. Then, the interaction sequences between students and exercises are processed and the exercise embedding vectors are spliced into the embedding vector module. Finally, the students' knowledge state module and the embedding module are utilized to calculate the probability of correctness of the next question using causally inflated convolution,which is then combined with the forgetting factor module to generate the final probability.

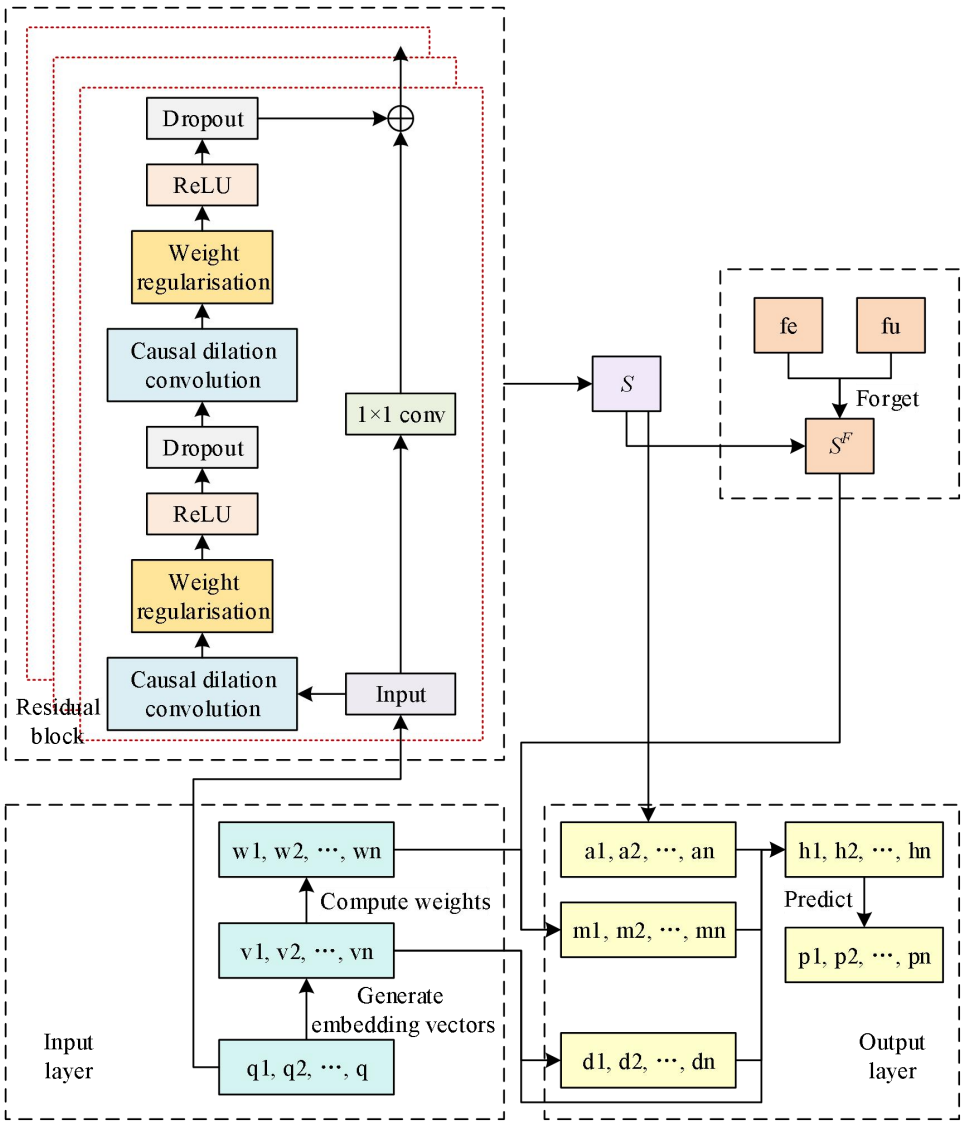

Figure 3. TCKT-FI model structure

2.2.1. Knowledge State Extraction Module

Convolutional networks deal with problems under one-dimensional convolutions, TCN wants to ensure that it can deal with temporal problems, so it uses causal inflated convolution, which protects future information data, is unidirectional, and ensures that the output at moment t is only dependent on inputs at moment t and its previous moments. The inflated convolution is able to increase the perceptual field of view without loss of information, so that each convolution will output a larger range of data.TCN can learn all the input data using as few convolutional layers as possible. However, in some cases, if you simply increase the depth of the network, you will not only have problems with gradient vanishing and gradient explosion, but also network degradation, so TCN adds residual connections. Each residual module contains two causally inflated convolutional layers and a nonlinear mapping of ReLU, while regularizing the network. Among them, the ReLU function and Dropout are applied to effectively prevent overfitting.

Define the knowledge mastery state matrix $S \in R^{N \times K}$, where N denotes the number of student interaction records, K denotes the number of topics, and $S_t \in R^K$, denotes the knowledge mastery of the student at the moment t.

2.2.2. Oblivion module

The module of forgetting processing mainly focuses on forgetting the state of knowledge mastery at the end of the last learning. According to the phenomenon of forgetting in learning and the law of

forgetting in pedagogy, four factors can be proposed for the forgetting behavior: the time interval of repeating the same knowledge point (RK), the time interval from the last learning (RL), the number of times of repeating the same knowledge point (KT), and the degree of mastery of the knowledge point (KM).

The four forgetting factors are combined together to obtain the forgetting matrix F_t, which represents the four factors of forgetting. The students' forgetting factors $F_t(i)$ for knowledge point i are converted into forgetting vector $fe_t(i)$ by a Sigmoid function. Then the students' forgetting factor $F_t(i)$ for knowledge point i is converted to update vector $fu_t(i)$ through a Tanh function as shown in Eqs. (14) and (15):

$$fe_t(i) = Sigmoid\left(FE^T F_t(i) + b_{fe}\right) \tag{14}$$

$$fu_t(i) = \tanh\left(FU^T F_t(i) + b_{fu}\right) \tag{15}$$

The knowledge mastery state matrix at the end of the last study is forgotten with the generated forgetting vector to simulate the knowledge mastery of the manager's forgetting behavior after a period of unlearning:

$$S_t^F = S_{t-1}(1 - fe_t)(1 + fu_t) \tag{16}$$

2.2.3. Embedded modules

q_{it} denotes the question answered by the student i at moment t, and a_{it} denotes the right or wrong answer result, represented by the value 1 or 0. Randomly initialize the embedding matrix $A \in R^{(M*K)}$, M denotes the number of exercises, and K denotes the embedding vector dimension. Multiply the exercises q_t with matrix A to get the exercise embedding vector v_t, then make inner product with the knowledge point embedding vector N_t, and process the result with Softmax function to get the weight vector w_t as shown in Eq. (17):

$$w_t(i) = Soft\max\left(v_t^T N_t(i)\right) \tag{17}$$

2.2.4. Learning and Prediction Module

Item Response Theory (IRT) is a common theory in cognitive psychology, usually an item is affected by the difficulty of the exercise and the student ability, in this paper, the difficulty of the exercise d_{t+1} is represented by Tanh function calculation. Student ability a_{t+1} can be calculated by the students' current knowledge level, which is calculated using the Tanh activation function. The formulas are shown in (18) and (19):

$$d_{t+1} = \tanh\left(W_D^T v_{t+1} + b_D\right) \tag{18}$$

$$a_{t+1} = \tanh\left(W_A^T S_{t+1} + b_A\right) \tag{19}$$

where W and b denote the weight vector and bias vector in the fully connected layer.

The knowledge point related weight vector w_{t+1} and the knowledge state matrix are weighted and summed to obtain the weighted mastery vector m_{t+1}:

$$m_{t+1} = \sum_{i=1}^{K} w_{t+1}(i) S_{t+1}^F(i) \tag{20}$$

Then the vector m_{t+1}, the vector v_{t+1}, the vector d_{t+1} and the vector a_{t+1} are combined to get the new vector $[m_{t+1}, v_{t+1}, d_{t+1}, a_{t+1}]$, and calculated with Tanh function to get y_{t+1}, and then input h_{t+1} into the Sigmoid function to get p_{t+1}, which denotes the predicted probability of the student answering the next question correctly as shown in Eqs. (21) and (22):

$$h_{t+1} = \tanh\left(\left[m_{t+1}, v_{t+1}, d_{t+1}, a_{t+1}\right]\right) \tag{21}$$

$$p_{t+1} = Sigmoid(h_{t+1}) \tag{22}$$

In this paper, the cross-entropy loss function L is used, where a_t denotes the true value and p_t denotes the predicted value, and the loss function L is shown in equation (23):

$$L = -\sum_{t=1}^{N} \left(a_t \log p_t + (1 - a_t) \log(1 - p_t) \right) \tag{23}$$

2.3. Deep Reinforcement Learning Personalized Recommendation Algorithm Based on User State Representation Modeling

Reinforcement learning recommendation framework DRR based on user state representation modeling is shown in Fig. 4, DRR framework mainly consists of three modules, which are Actor network, Critic network and user state representation learning module.

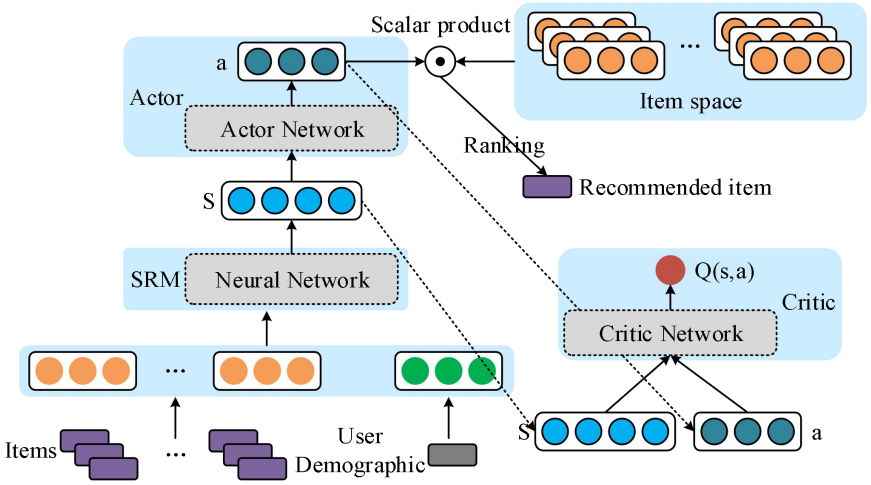

Figure 4. State representation modeling for deep reinforcement leaning

2.3.1. Actor network

The Actor network is also known as the policy network. For a user, the purpose of the Actor network is to generate an action a based on the current user's state s. The user's state s is obtained by its state representation learning module, which will be elaborated on later. Specifically, at time t, the user's state can be represented as:

$$\begin{aligned} s_t &= f(h_t) \\ h_t &= \{q_1, \ldots, q_n\} \end{aligned} \tag{24}$$

where $f(\cdot)$ denotes the user state representation learning module. Its input h_t denotes the set of embedding vectors of the user's positive browsing records in the most recent time slice, and $q_i \in \mathbb{R}^{1\times d} \left(i \in 1, \ldots, n\right)$ is a d-dimensional vector denoting the embedding vectors of the i th item. When the recommender system recommends an item $v \in \mathbb{V}$, if this user provides positive feedback, then at the next moment, the user's status is updated to:

$$\begin{aligned} s_{t+1} &= f\left(h_{t+1}\right) \\ h_{t+1} &= \{q_2, \ldots, q_n, q_v\} \end{aligned} \tag{25}$$

If not, $h_{t+1} = h_t$. Through a two-layer fully connected network, the user state s will be output by the Actor network to the action a, formally defined as:

$$a = \pi_\theta(s) \tag{26}$$

where π denotes the current recommendation strategy and θ denotes the parameters of the Actor network. It is worth noting in particular that the action a here does not represent an item to be

recommended, but a ranking function represented by a vector of continuous values, i.e., $a \in \mathbb{R}^{1\times d}$. The items in the candidate set can be ranked by the action a , and the ranking score for each item is defined as follows:

$$score_v = q_v a^T \tag{27}$$

Then, each item is sorted according to its score from highest to lowest, and the highest-ranked item is finally recommended to the user.

2.3.2. Critic Network

Critic network is also called value network.Critic network is actually a deep Q network which utilizes a multilayer neural network to estimate the true state-action pair value function $Q^{\pi}(s,a)$, referred to as Q value function. The function of this Q value function is to predict the Q value of the current action a , which is used to evaluate the merits of the current recommendation strategy. Specifically, the inputs to the Critic network are the user state s and the action a . Where s is generated by the user state representation module and a is the output of the Actor network.The output of the Critic network is the Q value of this state-action pair, i.e., $Q_{\omega}(s,a)$, which is a scalar. Based on this Q value, a temporal difference learning method can be used to optimize the recommendation strategy. In this case, the parameters of the Actor network are updated in the direction of increasing the Q value. According to the deterministic policy gradient theorem, the parameters of the Actor network can be updated by the sampled policy gradient:

$$\nabla_{\theta} J(\pi_{\theta}) \approx \frac{1}{N}\sum_{t} \nabla_a Q_{\omega}(s,a)\Big|_{s=s_t,a=\pi_{\theta}(s_t)} \nabla_{\theta}\pi_{\theta}(s)\Big|_{s=s_t} \tag{28}$$

where $J(\pi_{\theta})$ is the expectation of all possible Q values generated by the recommendation strategy π_{θ} . Here, N Q values are sampled by a small batch of strategies and then averaged to find their expectations, and the strategy gradient is found according to the chain rule. Also, the parameters of the Critic network can be updated by the time difference method, i.e., minimizing the mean square error:

$$L = \frac{1}{N}\sum_{t}\left(y_t - Q_{\omega}(s_t,a_t)\right)^2 \tag{29}$$

where y_t is the target Q value, which can be obtained through the Bellman equation:

$$y_t = r_t + \gamma Q_{\omega'}(s_{t+1}, \pi_{\theta'}(s_{t+1})) \tag{30}$$

where r_t is the immediate reward, the definition of the reward function will be described in detail subsequently, and γ is the discount factor. In the DRR recommendation framework, the goal network technique is employed to stabilize the training of the model. ω' and θ' are the parameters of the target Actor network and target Critic network, respectively.

2.3.3. State representation module

This section will first discuss how to explicitly utilize user-item interaction information to model user states. Specifically, four different network structures for modeling user state representation are shown in Fig. 5, with (b), (c), and (d) denoting the DRR-p, DRR-u, DRR-ave, and DRR-att user state representation structures, respectively.

(1) This section proposes a user state representation structure based on multiplicative networks, as shown in Fig. (a). The recommendation model based on this structure is named DRR-p because of its ability to capture the local dependency between two by two items through a per-bit multiplication operation. Moreover, in this multiplication operation, for each item a corresponding weight is learned to indicate the importance of each item for the state representation. Therefore, this state representation structure can be formally defined as:

$$\begin{aligned} s &= \left[h, \left\{e_{i,j} \,|\, i,j = 1,\ldots,n\right\}\right] \\ e_{i,j} &= w_i q_i \otimes w_j q_j \end{aligned} \tag{31}$$

where $\otimes$ denotes per-bit multiplication operations and w_i is a scalar to represent the importance of item v_i. The $e_{i,j}$ is a d-dimensional vector denoting the interaction of items v_i and v_j, and the dimension of the state s is $d(n+n(n-1)/2)$. From this structure, it can be observed that DRR-p, in addition to doing a per-bit multiplication operation on the two-two items to capture the information about the interaction between the two-two items, also clones these n items to keep their original information. Therefore, $n+n(n-1)/2$ feature vectors are eventually generated as the final state representation.

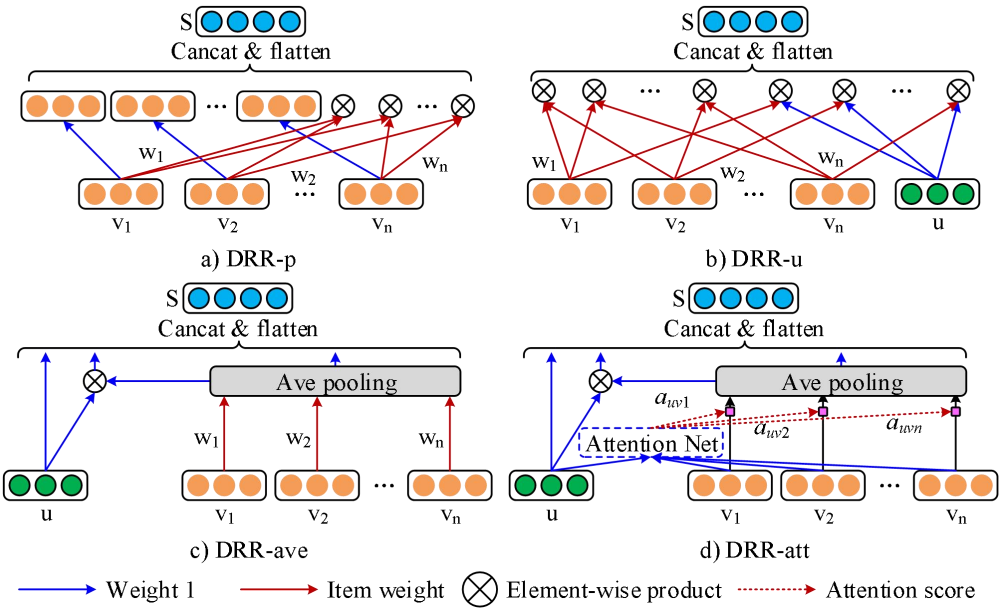

Figure 5. Four different user states represent the network structure modeled

(2) Although DRR-p can model the dependencies between two-two items, the interaction information between users and items is missing. To alleviate this situation, this section proposes the DRR-u structure, as shown in Fig. (b). From the DRR-u structure, it can be found that in addition to modeling the dependencies between two-by-two items, the interactions between users and items are also modeled through the per-bit multiplication operation. The user state can be formally represented as equation (32), and the dimension of the state s is also $d(n+n(n-1)/2)$.

$$s=\left[\left\{p_u \otimes w_i q_i \middle| i=1,\ldots,n\right\},\left\{e_{i,j} \middle| i,j=1,\ldots,n\right\}\right] \tag{32}$$

(3) In the DRR-p and DRR-u structures, the interactions between users and items are learned through bitwise multiplication operations. However, these two structures may cause higher computational complexity if the length of h is increased. In this section, we propose a network structure DRR-ave that incorporates the user-item interaction relationship, as shown in Fig. (c). From its structure, it can be found that in order to avoid computing the multiplication operation between two two embedding vectors, DRR-ave uses an average pooling layer with weights to first do fusion of items in h. Then, it utilizes the item vectors after fusion to do per-bit multiplication with the user vectors. Finally, DRR-ave connects this fused item vector, user vector, and the vector after interaction as the final state representation vector.The state s in DRR-ave can be formally defined as Equation (33), where the dimension of the state s is $3d$.

$$s=\left[p_u, p_u \otimes \left\{ave(w_i q_i) \middle| i=1,\ldots,n\right\},\left\{ave(w_i q_i) \middle| i=1,\ldots,n\right\}\right] \tag{33}$$

(4) Different users also have different preferences for the same item. To solve this problem, a new structure DRR-att is proposed in this section. where an attention network is used to generate item weights related to specific users. The attention network is a multilayer fully connected network with an activation function of ReLU. the inputs to the attention network are user vectors and item vectors, and the output layer is a softmax layer that is used to generate the weights of each item. For example, given

user u , for each item v in h , the attention network generates a weight a_{uv} . Then, an average pooling operation is done using that weight. This weight generation process can be formally defined in Eq. (34). Where $W_1 \in \mathbb{R}^{d_1 \times 1}$, $W_2 \in \mathbb{R}^{2d \times d_1}$ and $b_1 \in \mathbb{R}^1$, $b_2 \in \mathbb{R}^{1 \times d_1}$ are the weights and biases of this attention network. The final state representation obtained is still $3d$.

$$
\begin{aligned}
a'_{uv} &= \mathrm{Re}LU\left(\left(\left[p_u, q_v\right]W_2\right)+b_2\right)W_1+b_1 \\
a_{uv} &= \frac{\exp\left(a'_{uv}\right)}{\sum_{i=1}^{n}\exp\left(a'_{ui}\right)}
\end{aligned}
\tag{34}
$$

2.4. Model Training and Evaluation Process

2.4.1. Training process

The training algorithm consists of two main phases, the interaction generation process and the model update process. In the first stage, the recommender system observes the user's state s_t , and then obtains the action $a_t = \pi_\theta(s_t)$ through the policy network combined with the $\varepsilon - greedy$ exploration mechanism. After getting the current action, then the highest scoring item calculated according to the formula is recommended to the user. Next, the instant reward r_t is calculated based on the user's feedback and history information. Then, the user's status is updated. Finally, the above interaction record quaternion $\left(s_t, a_t, r_t, s_{t+1}\right)$ is stored into the experience playback buffer D .

In the second stage, a batch size of N interaction experience records are first randomly sampled from D . Then, the parameters in the Actor network and Critic network are updated according to Eq. Also the parameters in the state representation module update the parameters according to the gradient update signal in Actor. In addition, this section also employs the widely used goal network mechanism to smooth the training of the network and prevent the gradient dispersion during parameter update.

2.4.2. Offline assessment

For a given session record, the recommender system only recommends the items that appeared in the session rather than the entire candidate set, because there are only records of the items that have appeared in the historical record data, and it is not possible to give an accurate feedback on the items that have not appeared. Therefore, the offline evaluation process can be regarded as a reordering operation of the items in the session records according to the user's interests and preferences. After the whole recommendation process is finished, the evaluation index can be calculated by comparing the recommended sequence L and the true value sequence, and the model parameters are not updated in this process.

2.4.3. Incentive Shaping

The reward function $R(s,a)$ is a metric used to evaluate the quality of the current recommendation program, specifically, the reward function in this chapter is defined as:

$$
R(s,a) = R_0(s,a) + \alpha\phi(s,a) \tag{35}
$$

where $R_0(s,a)$ denotes the raw rewards, i.e., the immediate rewards are calculated based on the user feedback from the user's historical browsing history. Where all raw rewards are normalized to the $[-1,1]$ interval. $\phi(s,a)$ is the potential reward function, which can be considered as an objective function for local information.

3. Results and Discussion

3.1. Experiments to test the effectiveness of knowledge tracking

The experiments are based on four authoritative online education datasets, and compared and analyzed with the traditional model to demonstrate the impact of the TCKT-FI model proposed in this paper on students' learning status. The configuration environment used for the experiment:The processor is Intel i7 6700H CPU with 2.6GHz, 16GB RAM, and GTX 960 graphics card.The programming language used for the experiment is Python, which is implemented on the framework of

TensorFlow 1.16.0, which adopts Pytorch, Numpy, and Pandas, Matplotlib as dependent packages.

3.1.1. Introduction to the data set

Four real online education datasets were used for the experiment: assitments2009, assitments2012, assitments2015, and slepemapy.cz. The datasets are summarized as shown in Table 1.ASSITments2009 involves 128 different knowledge points, with a total of 325,652 records, completed by 4202 students: ASSITments2012 involves 252 knowledge points with 5818858 records completed by 45684 students. ASSITments2015 involves 106 knowledge points with 683772 records completed by 19864 students.slepemapy.cz is a geography online education system which includes 10087421 responses from 87948 students for 1446 knowledge point records.

Table 1. Dataset summary

Dataset	The number of students	Knowledge points	Total number of records
ASSITments2009	4202	128	325652
ASSITments20012	45684	252	5818858
ASSITments20015	19864	106	683772
Slepemapy.cz	87948	1446	10087421

3.1.2. Evaluation indexes and experimental parameter settings

In this chapter, in order to test the influence of forgetting and remembering factors on students' learning status and the accuracy of the TCKT-FI model, the performance of the traditional knowledge tracking model and the TCKT-FI model on four real data was determined. In this chapter, AUC, ACC, and RMSE are used as the indicators to evaluate the prediction performance, respectively.The higher the values of AUC and ACC, the more accurate the prediction performance is, and the lower the value of RMSE, the lower the error is and the more accurate the prediction performance is. As a model comparison, BKT, DKT, DKVMN and TCKT-FI models will be selected as the evaluation targets in this chapter.

1) Impact of training set size on models

In order to better compare the performance change state of the model under different training sets, the study sets different training set sizes respectively. The AUC values of this paper's algorithm in each training set with different training set shares are shown in Fig. 6. From the results of the figure, 60% of ASSITments2009, 70% of ASSITments2012, and 80% of ASSITments2015 are selected as the training set, 40%, 30%, and 20% of the remaining portion are used as the test set, and 40%, 30%, and 20% are used as the validation set. Comparative experiments were conducted by using 80% of the geo-type data slepemapy.cz as training set, 10% as test set and 10% as validation set.

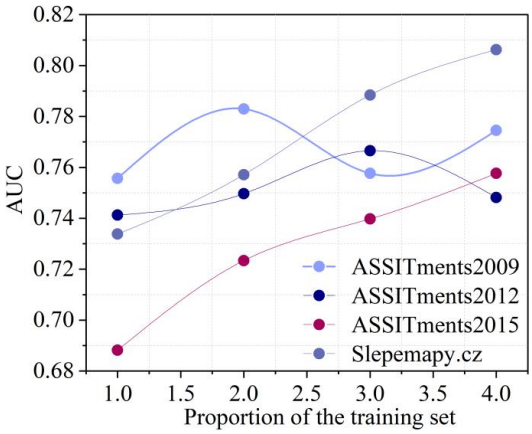

Figure 6. AUC values under different proportions of training sets

2) Influence of Knowledge Point Word Vector Dimension on Modeling

In order to study the effect of knowledge point word vector dimensions on the prediction results of knowledge tracking model, the knowledge point word vector dimensions were set to (8,16,32,64,128, 256), and the AUC values of TCKT-FI model in each dataset with different knowledge point word vector dimensions were tested, and the AUC values under different knowledge point word vector dimensions are shown in Fig. 7. The comparison shows that the AUC value of TCKT-FI model in

ASSITments2009, ASSITments2012, ASSITments2015, slepemapy.cz dataset is the highest when the knowledge point word vector dimensions are 128, 32, 128, and 256, which are: 0.83042, 0.80733, 0.83042, 0.81757.

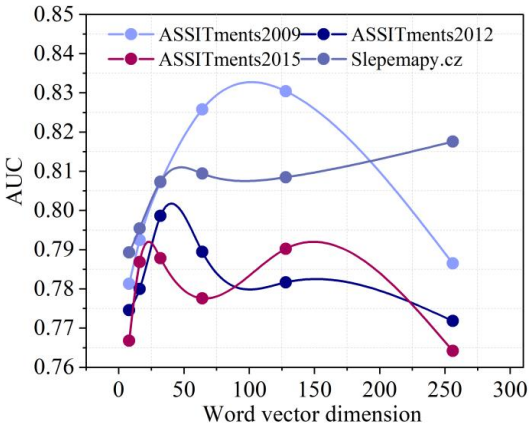

Figure 7. AUC values under different knowledge point vector dimensions

3.1.3. Predictive performance analysis

In this chapter, BKT, DKT, DKVMN are calculated by analyzing the model on ASSITments2009, ASSITments2012, ASSITments2015, slepemapy.cz dataset, for the real answer results of the ANSWERED field compared with the answer results predicted by the model, SATFKT and the result data of TCKT-FI model proposed in this paper on AUC, ACC, RMSE values.The result data of this paper's algorithm on AUC, ACC, RMSE values are shown in Table 2.The TCKT-FI model is used in the ASSITments2009, ASSITments2012, ASSITments2015, slepemapy.cz datasets with AUC values of 0.8504, 0.8117, 0.7938, 0.8304,ACC values of 0.7645, 0.7024, 0.6987, 0.7697, and RMSE values of 0.3855, 0.3749, 0.3797, 0.387, respectively, which is a very good result in comparison to the state-of-the-art DKVMN model on several datasets with an average AUC value of 3.21%.

Table 2. The results of the algorithm in AUC, ACC, RMSE values

	Model	**ASSITments 2009**	**ASSITments 2012**	**ASSITments 2015**	**Slepemapy .cz**
AUC	BKT	0.614	0.6734	0.7008	0.7019
	DKT	0.7877	0.7256	0.7402	0.787
	DKVMN	0.8301	0.7678	0.7676	0.7924
	SATFKT	0.8456	0.7909	0.788	0.8192
	This model	0.8504	0.8117	0.7938	0.8304
ACC	BKT	0.6957	0.6334	0.5794	0.6738
	DKT	0.7371	0.6781	0.6182	0.732
	DKVMN	0.7519	0.6879	0.6663	0.7405
	SATFKT	0.7526	0.6999	0.6872	0.7671
	This model	0.7645	0.7024	0.6987	0.7697
RMSE	BKT	0.439	0.4423	0.4195	0.4308
	DKT	0.4122	0.4132	0.4112	0.4102
	DKVMN	0.4098	0.4007	0.3972	0.4021
	SATFKT	0.3927	0.3938	0.388	0.3871
	This model	0.3855	0.3749	0.3797	0.387

3.1.4. Analysis of knowledge tracking results

In this section, student learning records from ASSITments2015 during the months of October through December were used, which included 1556 test questions and the corresponding five knowledge points. In order to better analyze the data, the students' answer records are represented by the tuple (k_t, a_t), where k_t denotes the knowledge point that the student learned at the time t, and the answer at the time t a_t, with 0 denoting incorrect answers and 1 denoting correct answers. Taking the knowledge point "rewriting English statements" as an example, we analyze the knowledge point mastery of students a and b in the first 15 moments. The results of the knowledge point mastery

analysis are shown in Figure 8. a student answered the questions about the knowledge point correctly at moment 0 and moment 5, then the TCKT-FI model and the DKVMN model simulate that the students' mastery of the knowledge point is improved, but the improvement is more obvious under the TCKT-FI model, which is due to the fact that the TCKT-FI model adds the memory reading layer, which strengthens the students' ability of pre-memorization for the knowledge point. The students' ability to memorize the knowledge points in the previous moments. However, in the following moments 1.1~5, 5.8-9.5, 10.5-15, students did not learn the knowledge point, then the TCKT-FI model showed a significant downward trend, indicating that students had forgotten the knowledge point, but the downward trend in the DKVMN model was smaller, which did not simulate the forgetting process of the students well. b The students answered the question about the knowledge point wrongly in moment 10, and the TCKT-FI model was more obvious, because the TCKT-FI model added the memory reading layer, which enhanced the students' ability to remember the knowledge point in the previous period. The TCKT-FI model and the DKVMN model both simulate the students' mastery of the knowledge point to be reduced, and the effect performance is the same.

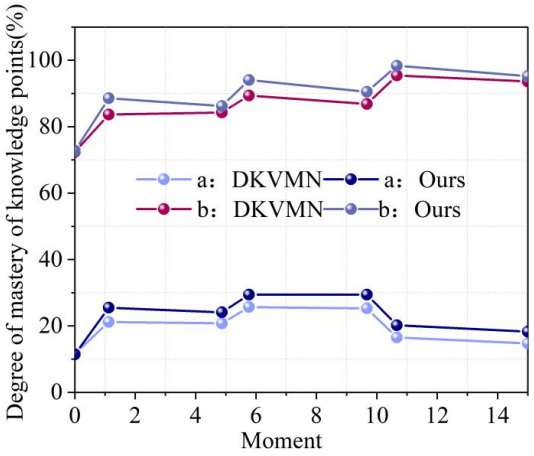

Figure 8. Analysis results of the mastery of knowledge points

The knowledge tracking outputs of this paper's algorithm and DKVMN in the ASSISTments2015 dataset are shown in Figures 9 and 10, respectively. As can be seen from the data in the figures, student a answered correctly to the test questions containing knowledge point k_3 at moments 1, 3, and 10, then both the TCKT-FI model and the DKVMN model simulated the growth of students' mastery of knowledge point k_3. At moment 5, when student a answered incorrectly to the question containing knowledge point k_3, both the TCKT-FI and DKVMN models simulated a decrease in the student's mastery of knowledge point k_3. However, the DKVMN model simulated that students' mastery of knowledge point k3 did not change significantly at other moments when students did not learn knowledge point k_3, but the TCKT-FI model simulated that students' mastery of knowledge point k_3 was slightly reduced, which represented students' forgetting of knowledge point k_3, and verified the correctness of the TCKT-FI model for the modeling of the process of students' forgetting and remembering.

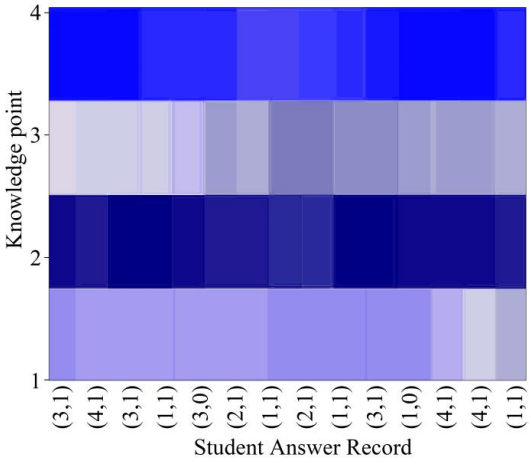

Figure 9. The knowledge of the algorithm is tracked

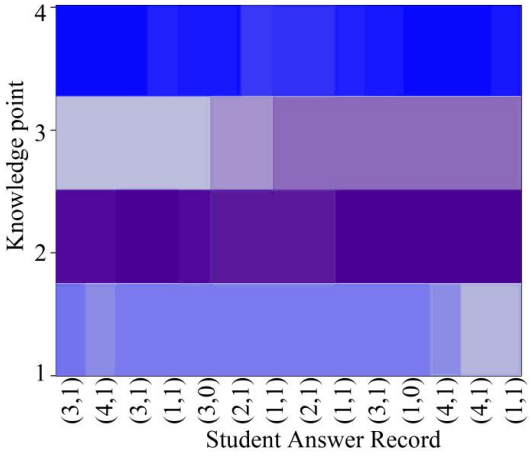

Figure 10. DKVMN knowledge tracking output results

3.2. Model performance testing experiments

3.2.1. Experimental setup

The experiments in this section are conducted based on an English writing knowledge point dataset. The comparison methods used in this experiment are Probability Matrix Method (PMF), Heuristic Diversity Method (MMR), Supervised Learning Approach (SUP-LSTM), Supervised Learning Approach (SUP-DIV), which is similar to SUP-LSTM, with the difference of using a recurrent bilinear model similar to the one used in MDP-DIV, Reinforcement Learning-based Diversity Approach (MDP-DIV), a combined model of LSTM's model and MDP-DIV's optimization method (MDP-LSTM), an asynchronous Action Advantage-based Actor-Critic algorithm (A3C-GAE), and an improved version of the Actor-Critic algorithm (AC-QSA).

In order to avoid the bias of evaluation metrics, two different evaluation metrics including aNDCG and ERRIA are used in this experiment. when the length of recommendation list in the experiment is K, the corresponding metrics are aNDCG@K and ERRIA@K.

3.2.2. Diversity

This paper first compares the final results of the proposed method in this paper and each comparison method on diverse recommendations. This experiment will implement different reward programs according to the metrics during training. The values of the evaluation metrics for each diversity recommendation method are shown in Table 3. For all the diversity evaluation metrics, PMF, which focuses only on relevance, performs the worst, followed by MMR, a heuristic diversity method. The supervised learning based methods SUP-LSTM and SUP-DIV show significant improvement over MMR. Note that here SUP-LSTM uses the same policy model as the present method, and the performance improvement here also indicates the effectiveness of the present method. Compared with SUP-DIV, SUP-LSTM converges faster, and similar performance can be found when comparing MDP-LSTM and MDP-DIV based on policy gradient.

Table 3. The evaluation index values of various diversified recommendation methods

Metric	aNDCG@1	aNDCG@5	aNDCG@10	ERRIA@1	ERRIA@5	ERRIA@10
PMF	0.6389	0.6613	0.6939	0.5839	0.608	0.6262
MMR	0.6404	0.7556	0.7804	0.5876	0.6426	0.6691
SUP-LSTM	0.9005	0.9132	0.9147	0.7056	0.7632	0.7843
SUP-DIV	0.9065	0.9099	0.9117	0.7151	0.7714	0.7861
MDP-DIV	0.7981	0.789	0.7941	0.6366	0.6984	0.7169
MDP-LSTM	0.7957	0.815	0.8182	0.6457	0.7145	0.7352
A3C-GAE	0.9034	0.8986	0.8949	0.7193	0.7699	0.7895
AC-QSA	0.9076	0.9141	0.9148	0.7186	0.7672	0.7874
Ours	0.9078	0.9196	0.922	0.7281	0.7872	0.8034

3.2.3. Learning efficiency

In this section, the training stability and convergence speed of each method are analyzed. The change curve of the evaluation metrics with the training cycle is shown in Fig. 11. The test metric here is aNDCG@10, and it is clear from the figure that the metric value of this paper's method is higher than the other methods after each method in convergence. Usually reinforcement learning algorithms need to be explored sufficiently to have a better action and start performing well. It can also be seen from the figure that the comparison method based on supervised learning outperforms the traditional reinforcement learning based method in the starting phase and converges quickly, but nevertheless, this paper's method stably outperforms the supervised learning method thanks to the efficient sample utilization efficiency based on the complete action space in this work.

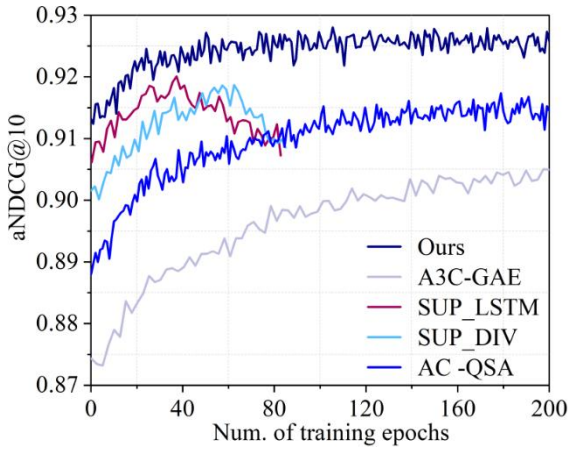

Figure 11. The variation curve of evaluation indicators with the training period

3.2.4. Exploring Strategies

Exploration mechanisms play an important role in reinforcement learning. In this experiment, we investigated the effects of three different exploration strategies, including attenuated $\varepsilon-greedy$, fixed $\varepsilon-greedy$ (ε=1), and ε=0. ε=1 and ε=0 correspond to strategies that sample actions that are completely randomized based on action probabilities, and purely greedy selection of optimal actions without exploration. A comparison of the training curves for the different exploration strategies is shown in Fig. 12. As can be seen from the figure, the decaying $\varepsilon-greedy$ and the completely randomized exploration strategy ε=1 are clearly due to the greedy strategy without exploration (ε=0). This shows the important impact of exploration strategy in reinforcement learning algorithms. It can also be seen that the decaying $\varepsilon-greedy$ also outperforms the completely randomized exploration strategy, which demonstrates the positive significance of employing different explorations and exploiting the balancing mechanism at different training stages.

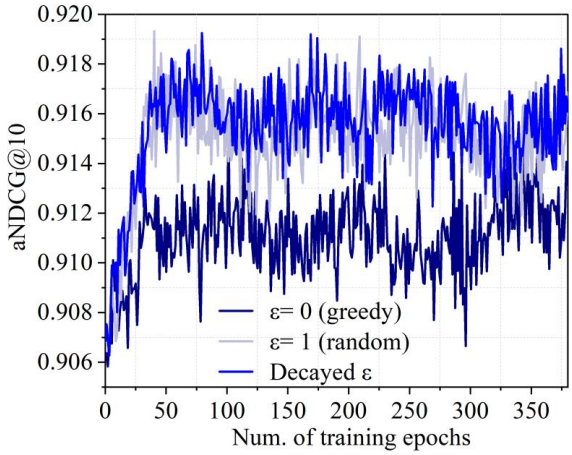

Figure 12. Comparison of training curves of different exploration strategies

3.2.5. Supervisory losses

In this section, the impact of the parameters related to the supervisory loss component will be analyzed, and the parameters considered here include λ, i.e., the weight of the supervisory loss, and τ, i.e., the differentiation threshold for the action Critic value. The impact of the values of the parameters related to the supervisory loss is shown in Table 4. It can be seen that the performance of λ =0.1 is superior to both λ =0.01 and λ =1 settings. In fact, as λ decreases to 0, the method proposed in this paper gradually approaches the pure reinforcement learning method, while as the in increases, the method will weaken the optimization effect of reinforcement learning and gradually approaches the pure pair-wise supervised learning method. For τ, from the right figure, we can see that τ =0 achieves the best result, which may be due to the fact that the greedily constructed supervised learning samples have a certain bias.

Table 4. Monitor the influence of parameter values related to losses

		aNDCG@5	**aNDCG@10**
λ	0.01	0.9172	0.9184
	0.1	0.9180	0.9195
	1	0.9168	0.9185
τ	0	0.9181	0.9195
	0.1	0.9170	0.9187
	0.5	0.9175	0.9189

3.2.6. Decay factor

This experiment investigates the effect of the attenuation factor γ based on the variation of the recommended location, consistent with the evaluation metrics, e.g., for aNDCG. Consistent with the evaluation metrics, such as for aNDCG. This experiment compares this diversification metric oriented γ setting with the traditional fixed attenuation rate scenarios. The impact of the values of the decay rate parameter for future earnings is shown in Fig. 13. Where for the fixed decay rate setting as the best performing γ = 0.88. As seen in the figure, the change discount rate used in this paper not only achieves higher metrics than the optimal fixed decay scheme, but also performs better on average especially in the later stages of training. This approach removes the burden of positional bias when estimating state values, and experiments validate its effectiveness.

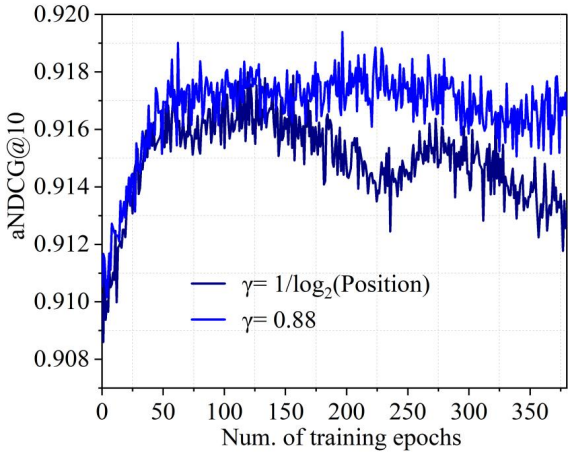

Figure 13. The influence of the parameter value of the future income attenuation rate

4. Conclusion

The article proposes a deep knowledge tracking model based on reinforcement learning theory, which enriches the semantic information that the model has and enhances the interpretability of the model. A deep reinforcement learning recommendation framework DRR based on Actor-Critic architecture is designed, and the article verifies the effectiveness of the proposed model through experiments, with the following specific conclusions:

(1) In the analysis of knowledge tracking results, take the knowledge point “English statement rewriting” as an example, and analyze the knowledge point mastery of students a and b in the first 15

moments. It is found that both the TCKT-FI model and the DKVMN model simulate that the students' mastery of the knowledge point is improved, but the improvement is more obvious under the TCKT-FI model, which is due to the fact that the TCKT-FI model adds a memory reading layer, which strengthens the students' ability to memorize the knowledge point in the early stage. In the subsequent moments, the model can well simulate the students' forgetting process, which shows that the model has a good effect of tracking English writing knowledge.

(2) In the evaluation index value experiments of diversified recommended methods, the performance of this paper's method is optimal compared with other methods, for example, under the aNDCG@1 index, the evaluation index value of this paper's method is the highest of 0.9078, which shows the effectiveness and superiority of this paper's method, and greatly improves the training stability and training efficiency compared with other reinforcement learning methods.

About the Author
Shali Jin (1990.9.12-), female (Hui ethnicity), born in Yinchuan, Ningxia, lecturer, Master's degree, research field: Education.

References

1. Jiang, L., Yu, S., Zhou, N., & Xu, Y. (2023). English writing instruction in Chinese students' experience: A survey study. RELC Journal, 54(1), 37-54.
2. Zhou, A., & He, H. (2025). English curriculum reform in China and teachers' preparedness to implement the latest curriculum standards. In Oxford Research Encyclopedia of Education.
3. Zhang, C., Yan, X., & Liu, X. (2015). The development of EFL writing instruction and research in China: An update from the International Conference on English Language Teaching. Journal of Second Language Writing, 30, 14-18.
4. Shilon, S. G. (2024). School Leaders as Boundary Spanners: Shared Sense-Making Processes for Personalized Learning within a National Curriculum Reform Implementation. In Personalization in Pedagogical Landscapes in the Digital Age-A Global Perspective. IntechOpen.
5. Mourtzis, D., Panopoulos, N., & Angelopoulos, J. (2023). A hybrid teaching factory model towards personalized education 4.0. International Journal of Computer Integrated Manufacturing, 36(12), 1739-1759.
6. Bhutoria, A. (2022). Personalized education and artificial intelligence in the United States, China, and India: A systematic review using a human-in-the-loop model. Computers and Education: Artificial Intelligence, 3, 100068.
7. Hu, Z. (2025). Data-driven innovative models of personalized teaching. Discover Education, 4(1), 364.
8. Li, L. (2023). Classroom Teaching Decision-Making Optimization for Students' Personalized Learning Needs. International Journal of Emerging Technologies in Learning, 18(9).
9. Wu, S., Cao, Y., Cui, J., Li, R., Qian, H., Jiang, B., & Zhang, W. (2024). A comprehensive exploration of personalized learning in smart education: From student modeling to personalized recommendations. arXiv preprint arXiv:2402.01666.
10. Zhou, Z., & Li, W. (2024). Personalized college English learning based on artificial intelligence: Algorithm-driven adaptive learning method. International Journal of High Speed Electronics and Systems, 2540102.
11. Wang, F. (2023). IoT for smart English education: AI-based personalised learning resource recommendation algorithm. International Journal of Computer Applications in Technology, 71(3), 200-207.
12. Yuan, S. (2024). Personalized college english learning experience assisted by artificial intelligence: an algorithm-driven adaptive learning approach. International Journal of High Speed Electronics and Systems, 2540157.

13. Zhan, L. (2024, December). A Personalized English Teaching Model Based on Big Data and Intelligent Platform Empowerment. In Proceedings of the 2024 International Conference on Big Data Mining and Information Processing (pp. 273-278).
14. Dasam, S., Goriparthi, P., Manchanda, M., Nowbattula, P. K., & Subhashini, R. (2024, December). Enhancing writing skills with AI: Personalized feedback mechanisms for English learners. In 2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA) (pp. 1-6). IEEE.
15. Begum, S., Vimochana, M., Arunadevi, V., Dhivya, A. C. A., Vuyyuru, V. A., & Faizal, M. M. (2024, December). Using AI for Personalized English Writing Feedback on Educational Platforms. In 2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA) (pp. 1-6). IEEE.
16. Hwang, W. Y., Nurtantyana, R., Purba, S. W. D., Hariyanti, U., Indrihapsari, Y., & Surjono, H. D. (2023). AI and recognition technologies to facilitate English as foreign language writing for supporting personalization and contextualization in authentic contexts. Journal of educational computing research, 61(5), 1008-1035.
17. Palacios-Núñez, M. L., Mendoza-García, E. M., Zarate, J. W. N., & Deroncele-Acosta, A. (2025). ChatGPT in the Teaching of Academic Writing in Higher Education: Teachers' Perspectives on Its Uses, Challenges, and Future in Personalized Learning. Edutec, Revista Electrónica de Tecnología Educativa, (93), 33-50.
18. Zhu, J., Liao, Y., & Lu, D. (2024). Design and optimization of personalized education system based on intelligent algorithms. Procedia Computer Science, 243, 514-522.
19. Shakya, A. K., Pillai, G., & Chakrabarty, S. (2023). Reinforcement learning algorithms: A brief survey. Expert Systems with Applications, 231, 120495.
20. Den Hengst, F., Grua, E. M., el Hassouni, A., & Hoogendoorn, M. (2020). Reinforcement learning for personalization: A systematic literature review. Data Science, 3(2), 107-147.
21. Taylor, R., Fakhimi, M., Ioannou, A., & Spanaki, K. (2025). Personalized learning in education: a machine learning and simulation approach. Benchmarking: An international journal, 32(7), 2662-2689.
22. Tang, X., Chen, Y., Li, X., Liu, J., & Ying, Z. (2019). A reinforcement learning approach to personalized learning recommendation systems. British Journal of Mathematical and Statistical Psychology, 72(1), 108-135.
23. Shawky, D., & Badawi, A. (2018). Towards a personalized learning experience using reinforcement learning. In Machine learning paradigms: Theory and application (pp. 169-187). Cham: Springer International Publishing.
24. Sharif, M., & Uckelmann, D. (2024). Multi-modal LA in personalized education using deep reinforcement learning based approach. Ieee Access, 12, 54049-54065.
25. Amin, S., Uddin, M. I., Alarood, A. A., Mashwani, W. K., Alzahrani, A., & Alzahrani, A. O. (2023). Smart E-learning framework for personalized adaptive learning and sequential path recommendations using reinforcement learning. IEEe Access, 11, 89769-89790.
26. Su, Y. (2025). Personalized recommendation using deep reinforcement learning for educational content. International Journal of Image and Graphics, 2750029.