ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

Synergistic Development of Artificial Intelligence Algorithms for Precision Civics in Cybermimetic Environments

Wei Zhang[1,*]

[1] School of Accountancy, Sichuan Vocational College of Finance and Economics, Chengdu, Sichuan, 610101, China

SUMMARY: *In the era of artificial intelligence, algorithmic recommendation technology has reshaped the information dissemination pattern, promoted structural changes in the communication paradigm, and brought new contextual challenges to ideological and political education. The study proposes a knowledge tracking model F-TCKT that integrates the forgetting factor and the attention mechanism, uses TCN to process the historical interaction information of students' ideological and political learning, and integrates the information of sequential features of different sizes through the attention mechanism to realize the modeling of students' ideological and political level. On this basis, the recommendation process of Civics resources is transformed into a Markov decision process and modeled using a dual DQN model to improve the recommendation accuracy of Civics resources.The F-TCKT model improves the most on the ASSISTments2015 dataset, with the AUC value and the Acc value improved by 19.37 and 10.97 percentage points compared with DKT. At the same time, the recommendation effect gradually improves and stabilizes with the number of trainings, which can recommend the Civics resources according to the answering state at a certain moment, and realize the rapid improvement of students' Civics cognitive ability in a shorter period of time.*

KEYWORDS: *attentional mechanism; Markov decision making; dual DQN model; F-TCKT model; Precision Civics*

1 Introduction

With the prevalence of Internet information technology, the development of mimetic environments has seen transformative progress, resulting in network mimetic environments. At this stage, the number of Internet users is increasing, the network environment is getting more and more intense, and the network mimetic environment presents many new features and trends different from the traditional environment, such as diversification of the constructing body, fragmentation of communication media, personalization of the network audience, and weak feedback regulation, etc. When the ideological education and ideological work encounters the network mimetic environment in this field, the problem becomes even more problematic and complex [1-5].

In recent years, the booming development of Artificial Intelligence (AI) technology has fostered a new research paradigm mainly powered by arithmetic power. Among them, AI algorithmic technology manipulates the form and content of information sent and received by humans, reshaping their means of information transmission and creating a strong algorithmic power [6]. “AI algorithm + education” is a new trend in the development of education.AI

*zwei202407@163.com
https://doi.org/10.65102/is2026334 Published: 30 April, 2026

algorithms have the characteristics of fast, efficient and accurate, which is the highlight of the development of AI, and is valued by all sectors of society, especially the digital transformation of the education field and the realization of accurate and personalized teaching mode [7-9]. AI algorithmic power affects people's thinking behavioral habits and the shaping of correct values, the ideological educators should give full play to the advantages of algorithmic technology, coupling the ideological content in the learning and life of college students, and empowering the precise ideology [10, 11].

Precision teaching emerged in the 1960s, but due to the cumbersome operation, complex records, and the lack of uniform measurement standards, precision teaching has not been promoted on a large scale [12, 13]. The AI era has re-given life to precision teaching, and precision Civics has gradually become a hotspot of research on the reform of Civics course teaching, which puts forward new requirements for Civics courses in terms of both quality and efficacy [14-16]. How to give full play to the technical advantages of AI algorithms under the network mimetic environment, boost the effectiveness of precision Civics, optimize the process of Civics education in colleges and universities, realize the deep intermingling of Civics education and intelligent algorithms, and accurately satisfy the new needs of the education object, is an important topic that must be solved.

In this paper, an intelligent AI-based recommendation model for Civics teaching is constructed in a network mimetic environment. First, a deep knowledge tracking model F-TCKT incorporating forgetting factors is proposed to update students' knowledge point mastery and forgetting level in real time, and to predict students' future learning performance based on this. Then a recommender system algorithm DQNs based on two intelligences is proposed, which models the recommendation process as two Markov decision-making processes based on users and based on user groups, and uses DQNs in deep reinforcement learning to model them separately, aiming at recommending personalized Civics resources to students with different Civics levels. Finally, comparative experiments are conducted on four datasets, ASSISTments2009, ASSISTments2015, Statics2011 and Synthetic-5, to test the performance of the knowledge tracking model and the accurate Civics recommendation model, respectively.

2 Intelligent and Accurate Civics Teaching Model in Network Mimetic Environment

2.1 Enhanced learning

2.1.1 Markov decision-making process

Markov Decision Process (MDP) is the mathematically ideal form of reinforcement learning, which is based on the idea that an intelligent body is made to be in a state S_t at moment t, and after selecting action a_t, the environment state changes to S_{t+1} while a delayed reward is obtained R_{t+1}, after which the individual can continue to select the next appropriate action, generating a new environment state and a new reward value.

The basic framework of reinforcement learning is shown in Fig. 1, and the ultimate goal of reinforcement learning is to maximize cumulative gains by generating good strategies π through interaction with the environment.

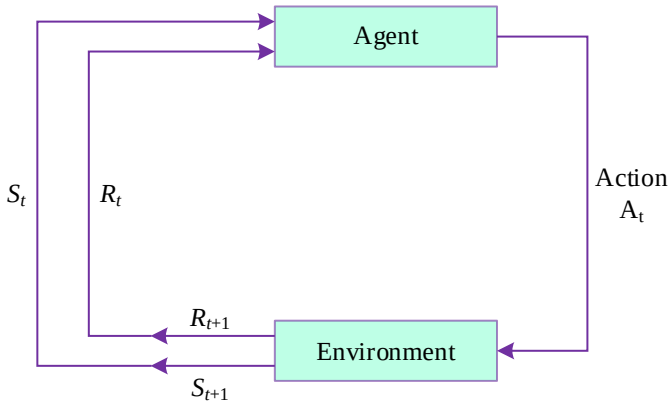


Figure 1: Strengthening learning basic framework

The Markov decision process can be represented as a quintuple $\langle A, S, P, R, \gamma \rangle$, which is described in Table 1.

Table 1: Markov decision model element

Element	Representation	t Moment Symbol Representation	Description
S	Finite set of states	S_t ,and $S_t \in S$ 1,	Current environmental conditions
A	Finite action set	a_t ,and $a_t \in A$	Actions in the current environment
P	Transition probability	$P_{SS'}^{a} = P\left[S_{t+1} = S' \middle\| S_t = s, A_t = a\right]$	The probability of the state transitioning from S_t to S_{t+1}
R	Return function	$R_s^a = E\left[R_{t+1} \middle\| S_t = s, A_t = a\right]$	The reward value of action S_t under state A_t
γ	Discount factor	γ , $\gamma \in [0,1]$	Adjust the proportion of current and future returns

Thus, the process of reinforcement learning interacting with the environment reduces to the process of continuously accumulating returns in a Markov decision process. Let G_t be the sum of all the returns from the start to the end state after t moments after decaying by a certain percentage, denoted by Eq:

$$G_t = R_{t+1} + \gamma R_{t+2} = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{1}$$

As γ approaches 0, the intelligence focuses on short-term gains and does not consider the impact of future returns on the present. As γ approaches 1, the intelligence focuses on long-term gains and future returns are as important as current returns.

2.1.2 Basic algorithms for reinforcement learning

In a Markov decision process, the objective of the intelligent body is to maximize the expected value of the cumulative gain. The value function, $v_\pi(s)$ which utilizes the expectation of gain

to measure the performance of the intelligent body in a given state, can be expressed as:

$$v_{\pi}(s) = E_{\pi}\left(G_t \middle| S_t = s\right) \tag{2}$$

The optimal value function is defined as:

$$v_*(s) = \max_{\pi} v_{\pi}(s) \tag{3}$$

or recursively defined as:

$$v_*(s) = \max_{a \in A(s)} \sum_{s'} P_{ss'}^{a}\left(R_{ss'}^{a} + \gamma V^*\left(s'\right)\right) \ \forall s \in S \tag{4}$$

The action value function $q_{\pi}(s,a)$ is expressed as the optimal value of taking action a for state s under policy π and can be expressed as:

$$Q_{\pi}(s,a) = E_{\pi}\left[G_t \middle| S_t = s, A_t = a\right] \tag{5}$$

The optimal action value function is defined as:

$$Q_*(s,a) = \max_{\pi} Q_{\pi}(s,a) \tag{6}$$

or recursively defined as:

$$Q_*(s,a) = \sum_{s'} P_{ss'}^{a}\left[R_{ss'}^{a} + \gamma \max_{a' \in A(s')} Q_*\left(s', a'\right)\right] \forall s \in S, a \in A(s) \tag{7}$$

The above equation is also known as the optimal Bellman equation, from which the optimal policy is obtained as:

$$\pi^* = \arg\max_{a \in A} Q_{\pi}(s,a) \tag{8}$$

In fact, if the model of the system, i.e., the transfer probability P and the reward function R, is known, the problem is a typical planning problem and can be solved using dynamic programming methods (DP), which specifically include value iteration methods, policy iteration methods, and so on. However, in the actual environment, very often we do not know the model, in which case the classical algorithms chosen are Monte Carlo methods, time difference methods, Q-learning methods, and so on.

2.1.3 Deep Q-networks

DQN uses a neural network approximation to replace the Q-value table in the original Q-learning algorithm, thus avoiding the limitations of the storage structure.Q-learning updates the Q-value table by continuous learning until it converges, but in real life the Q-value table is often incomplete due to the infinity of the states, thus a function with parameters is used instead of a Q-value table, outputting the Q-values of all the actions, which allows to fit the more complex problems.

DQN uses an empirical playback mechanism during training. Due to the nature of

reinforcement learning, the correlation between the training data is increased, which is prone to cause instability in the trained neural network. Thus, DQN creates a fixed-capacity experience pool to store the training samples (s_t, a_t, r_t, s_{t+1}) and extracts a certain number of samples each time to train the neural network, which increases the usability of the data while decreasing the correlation between the data.

The DQN adds a target Q network to reduce the correlation of the data. Two neural networks with the same structure are built, the current Q network is denoted by $Q(s,a;\theta)$ and the target Q network is denoted by $Q(s',a';\theta^-)$, where θ and θ^- denote the neural network parameters, and the parameters of the Q network are copied to the target network at every L time step, i.e., $\theta^- \leftarrow \theta$ and after that, θ^- they are kept constant for a fixed period of time. The current Q value function is used to approximate the target value function. The error function $L(\theta)$ between the target Q network and the current Q network in the environment ε can be expressed as:

$$L(\theta) = E\left[\left(Y - Q(s,a;\theta)\right)^2\right] \tag{9}$$

$$Y = r + \gamma \max_{a'} Q\left(s',a';\theta^-\right) \tag{10}$$

where Y is denoted as the target network and the network parameters are updated by minimizing the error between the target network Q values and the current network Q values. The objective of deep reinforcement learning depends on the network weights θ, which are derived in order to minimize the loss function:

$$\nabla_\theta L(\theta) = \left(r + \gamma \max_{a'} Q\left(x',a';\theta^-\right) - Q\left(x,a;\theta\right)\right)\nabla_\theta Q\left(x,a;\theta\right) \tag{11}$$

2.2 Knowledge tracking of students' civic level based on the F-TCKT model

The deep knowledge tracking model F-TCKT based on TCN neural network and attention mechanism proposed in this paper, the structure of F-TCKT network model is shown in Fig. 2.

The attitude of teenagers towards traditional culture is contradictory. They emotionally recognize it but fail to act accordingly. 67.3% of the students believe that traditional culture is meaningful and should be understood, yet they do not put it into practice. Only less than one-third of the students will actively pay attention to relevant cultural information and the proportion of those who have participated in traditional culture activities is only 24.6%. This indicates that their attitude towards traditional culture does not match their behavior. The dissemination of traditional culture has failed to effectively motivate teenagers to actively participate.

This article uses a multi-dimensional measurement model to evaluate acceptance, covering three levels: cognitive acceptance, emotional acceptance, and behavioral acceptance. The comprehensive acceptance calculation formula is: The F-TCKT model is divided into the input layer, the forgetting layer, the concatenation layer, the learning layer, and the output layer. The input layer takes the three factors that affect forgetting during students' learning process RTI_t, STI_t, LT_t, as well as students' exercises Q_t and answer results A_t as inputs; the forgetting layer uses a fully connected network to model the three factors that affect forgetting in students'

learning process RTI_t, STI_t, LT_t and obtain a vector representing the degree of forgetting during students' learning process KL_t; the concatenation layer concatenates the output of the forgetting layer KL_t and the student's answer sequence matrix X_t (including students' exercises Q_t and answer results A_t) to obtain the student's answer sequence matrix under the influence of forgetting factors $X_{t'}$, as the input of the learning layer; the learning layer updates the student's answer sequence matrix $X_{t'}$ of this learning through TCN and attention mechanism to Y_t, adaptively determining the importance of forgetting information and short-term feature information in the student's answer sequence matrix; the output layer takes Y_t as input and outputs a vector *value* representing the predicted student's knowledge level to indicate the student's mastery of the current knowledge point at the current moment.

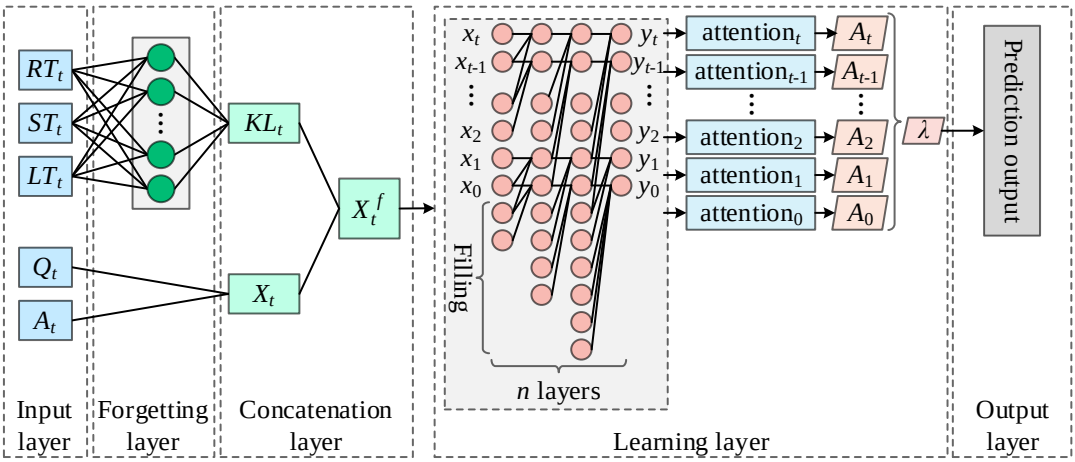


Figure 2: Structure of F-TCKT model

2.2.1 Input layer

Inputs are vector representations of each exercise under each knowledge point obtained from students' historical learning interaction records. The given student's practice process mainly contains exercise vector Q_t and answer result vector A_t. In previous knowledge tracking methods, Q_t and A_t are generally spliced as input vectors. In this paper the second input exists due to the introduction of factors affecting forgetting in the students' learning process. Organized by the theory of educational psychology and the data of the dataset, the given factors affecting forgetting in the process of students' learning mainly contain the time interval from the last time of learning the same knowledge point RTI_t, the time interval from the last time of learning STI_t, and the number of times of learning for the repetitions of the knowledge point LT_t.

2.2.2 Oblivion layer

The forgetting layer models the three factors affecting students' forgetting in each exercise under each knowledge point RTI_t, STI_t and LT_t, and outputs a vector representing the degree of students' forgetting in the exercise under the knowledge point KL_t. In this paper, we use a fully connected network to model the factors affecting forgetting, which gives up the good interpretability of perceptual machines and introduces a nonlinear activation function to

increase the model's representational ability, which is more like a composite of three-layer multiple logistic regression in terms of mathematical perspective. Mathematically, the fully connected network used in this paper is more like a composite of three-layer multivariate logistic regression. The three input vectors RTI_t, STI_t and LT_t are initialized with weights and biases, and trained with the main model to get the vector KL_t representing the degree of forgetfulness of the students, and the matrix form of the forgetfulness layer model is expressed as follows:

$$KL_t = W^2\left[\sigma\left(W^1 x_t + b^1\right)\right] + b^2 \tag{12}$$

$$\sigma = relu(x) = \max\{0, x\} \tag{13}$$

where: W^1 and b^1 are the weights and biases computed for the first and second tier networks; W^2 and b^2 are the weights and biases computed for the second and third tier networks; and x_t are the three factors affecting student forgetfulness RTI_t, STI_t, and LT_t.

2.2.3 Splice layer

The splicing layer is designed to combine the vector KL_t, which represents the degree of students' forgetfulness, output from the forgetting layer, with the matrix X_t of students' answer sequences (which includes the students' exercises Q_t and answer results A_t), to form a new matrix $X_{t'}$ of students' answer sequences containing the forgetting factors, which serves as the input to the subsequent main model TCKT.

After first obtaining matrix X_t, which represents the student answer sequence, and KL_t, which corresponds to the corresponding question under the corresponding knowledge point, the student's knowledge state at this time is modeled and represented by matrix $X_{t'}$ as:

$$X_{t'} = \left[X_t \oplus KL_t\right] \tag{14}$$

where: X_t is a matrix representing the sequence of student answers; KL_t is a vector of student forgetting for the corresponding question under the corresponding knowledge point.

2.2.4 Learning layer

The learning layer tracks the changes in knowledge mastery during the learning process based on the students' answer results, and models the students' learning behavior by updating the students' learning state matrix $X_{t'}$ to Y_t for this study through the TCN and attention mechanism.

In this paper, we have utilized the computational methods such as causal convolution, dilation convolution and residual joining in TCN, and then processed the TCN outputs through the attention layer to update the knowledge status of the students.

Causal convolution strictly follows the idea that predictions can only rely on the temporal information prior to the current moment, i.e., predictions can only be computed from input x_t at the current moment and previous inputs x_1 through x_{t-1}. Thus causal convolution allows strong causality in the model and also solves the problem of information leakage. The

conditional probability formula for causal convolution can be expressed as:

$$p(x)=\prod_{t=1}^{T} p\left(x_t \middle| x_1,\cdots,x_{t-1}\right) \tag{15}$$

where: x_t is the input at the current moment; $x_1,\cdots,x_{t-1}$ is the input before that moment. The formula can be abstracted as predicting the predicted probability of input x_t at the current moment based on the predicted values of $x_1,\cdots,x_t$ and for $x_1,\cdots,x_{t-1}$ and making it close to the actual value.

The structure of causal convolution ensures the causality of the time series data, but when the data length is long enough, the convolutional network needs more layers to be stacked in order to satisfy the requirement of causal convolution on the length of the data, so TCN introduces dilated convolution, which increases the range of the sensory field by fewer layers to be stacked. For a one-dimensional time series input, the computation of dilation convolution is defined as:

$$F(x)=(x*_d f)(s)=\sum_{i=0}^{k-1} f(i)\cdot x_{s-d\cdot i} \tag{16}$$

where: d is the dilation coefficient; k is the size of the convolution kernel; $x_{s-d\cdot i}$ is the size of the sensory field after the introduction of the dilation coefficient. In TCN, due to the deep number of convolutional layers, it is easy to bring the gradient problem, so the residual connection is applied instead of the connection between layers to enhance the generalization ability of the model. The use of residual connections is necessary when the network structure is deep. The residual connection is calculated as follows:

$$o=activation\left(x+\xi(x)\right) \tag{17}$$

where: x is the direct mapping part; $\xi(x)$ is the residual part and both are added together as the output of that part.The behavioral modeling of TCN for the student's learning state matrix $X_{t'}$ for this study can be expressed as:

$$Y_t'=TCN\left(X_1',X_2',\cdots,X_t';\theta\right) \tag{18}$$

where: X_t' is the input matrix at moment t; θ is the model parameters.

Considering the introduction of a deeper stack of residual connections, an attention mechanism is introduced into the model to alleviate the complexity and at the same time improve the computational efficiency of the model. In this paper, the TCN and the attention layer are viewed as a whole, and the soft attention mechanism is used to process the Y_t' matrix after the TCN update, which means that the whole learning layer updates the input student learning state matrix X_t^f to Y_t.

The attention distribution in the attention layer is calculated as:

$$key=value=Y_t^f \tag{19}$$

Using the scaled dot product model as the scoring mechanism, the attention layer is

calculated as follows:

$$attention\left(Q,K,V\right)=soft\max\left(\frac{Q\cdot K^{T}}{\sqrt{d_{k}}}\right)\cdot V \tag{20}$$

where: d_k is the last dimension of the Q-vector; the results of the attention layer are normalized by the $soft\max$-function, which outputs a probability distribution.

2.2.5 Predicting output and model training

The output layer takes as input the matrix Y_t, which represents the state of students' knowledge updated by the learning layer, and then processes the matrix Y_t by the fully connected layer, and outputs the vector $value$, which represents the predicted level of students' knowledge, and calculates the difference between the predicted value and the true value by minimizing the loss function, so as to optimize each parameter in the model. In this paper, BCELoss (binary cross entropy loss) loss function is used to optimize each parameter and Adam optimizer is chosen to train the model, the formula of BCELoss function is:

$$L=-\frac{1}{n}\sum\left(y\ln p(y)+(1-y)\ln(1-p(y))\right) \tag{21}$$

where: y is the category of the input value; $p(y)$ is the predicted value of that input value and is a probability value.

2.3 Recommendation of Civic Dynamics Based on Dual DQN Intelligentsia

2.3.1 Modeling the MDP of the student population

The Markov decision process for a group of users, $MDP(G)$ represents a personalized recommendation aimed at users, which is modeled based on the characteristics of that group of users, and the quintuple of $MDP(G)$ is defined as follows:

(1) State space: state s_t can be defined as the user's contextual information and the favorite hot content of this type of user, i.e., $s_t=\left(user,items\right)$. where $user$ denotes the user characteristics and $items$ denotes the set of hot content vectors.

(2) Action space: its definition is the same as $MDP(P)$.

(3) Reward function: unlike $MDP(P)$, the feedback value of $MDP(G)$ needs to show whether the recommended content is in line with the trend of the group, which can utilize the knowledge of statistics to count the number of user plays and likes as the feedback value.

(4) Transfer probability: when the user's contextual information, or the group's hot content changes, the state will be transferred.

(5) Discount factor: its definition is the same as $MDP(P)$.

2.3.2 Interactive Recommender System Based on Dual DQN Networks

In the previous subsection, we defined two MDP decision processes, and based on their respective roles, two value function structures need to be designed to store their respective Q values. The first is the local value function, which records the user's browsing behavior and

updates it based on individual user feedback. The second is the global value function, which maintains the changes of the whole system and tries to update it based on the feedback of the group of users.

The local value function needs to capture changes in the dynamic interests of users. In a user cold-start environment, the intelligent body does not have an initial state at the beginning, i.e., $s_t = \varnothing$. As the intelligent body continues to interact with the user, it slowly accumulates user interests for the purpose of personalized recommendation. Given the good performance of dynamic RNNs in dealing with such variable-length sequences, in this paper, a dynamic GRU is chosen to capture the user's interest at the moment of t. The GRU is computed in the same way as used in the previous two chapters. the input of the GRU is a low-dimensional dense vector of all the user's clicks in the sequence of $\{i_1, i_2, \cdots, i_N\}$, and, as in the previous work, the user's interest is expressed as 6 using the final hidden state of h_n as the output, i.e., the user's interest is expressed as $s_l = h_n$ in a local value function. the local value function uses a simple MLP for personalized recommendations. value function is nonlinearly approximated using a simple MLP. After obtaining the user's interest vector, the current local Q_l value is then output via a network of nonlinear value functions. It is expressed by Eq:

$$Q_l = f_{\theta_\pi}(s_l) \tag{22}$$

Here MLP is chosen as the parameter generating function with θ_π as its parameter. The local value function uses a two-layer neural network with an activation function of Relu as the hidden layer, and the output layer is linearly activated to obtain the Q value.

The global value function needs to capture the trend of the whole system. It needs to model information about the user's contextual environment, information about the user's own characteristics, and information about the popular content of the group of users to which the user belongs. These contents do not change as fast as the user's browsing behavior, so they are basically not dynamic, and their states are updated slowly, so special neural networks are not considered for modeling here. Let the state about the global value function be s_g and its output be Q_g, which is expressed by Eq:

$$Q_g = f_{\theta_\varphi}(s_g) \tag{23}$$

θ_φ are the parameters of the global value function, and here the simple MLP is chosen as the nonlinear function. The first two layers of the value function use Relu as the activation function and the value function uses a linear activation function to output Q values.

The recommender system aims to recommend more satisfactory goods to the user, and the goods recommended to the user at each moment are determined by multiple factors, and from the redefined MDP process, it can be seen that two value functions can be obtained about the respective MDP process, and next, it is necessary to recommend the action a_t based on the Q value at the moment of t. Based on the state of the user at the moment of t, the Q_l denotes the Q value of a local value function, which represents the personalized user's own preferences; based on the state of the user's group at t the moment, Q_g denotes a global Q value representing the trend of the user group. Determined by the two Q values together, we use a weighted sum to balance them to a predicted Q_{total} value:

$$Q_{total} = \omega Q_l + (1-\omega) Q_g \tag{24}$$

where ω is the weighting factor. In this way, the Q_{total} -value of each item in the candidate set can be obtained.In order to cope with the cold start problem, the intelligent body also needs to explore the user preferences, so the common $\varepsilon - greedy$ -exploration strategy is used for action selection. That is, when the random number is larger than ε, the action with the largest Q_{total} value is output, and when the random number is smaller than ε, a random strategy is used to select an action.

Based on the value function, the classical DQN intelligences are used to interact with the user, and the model framework proposed in this chapter is shown in Fig. 3. Two of the intelligences have their own experience pools.

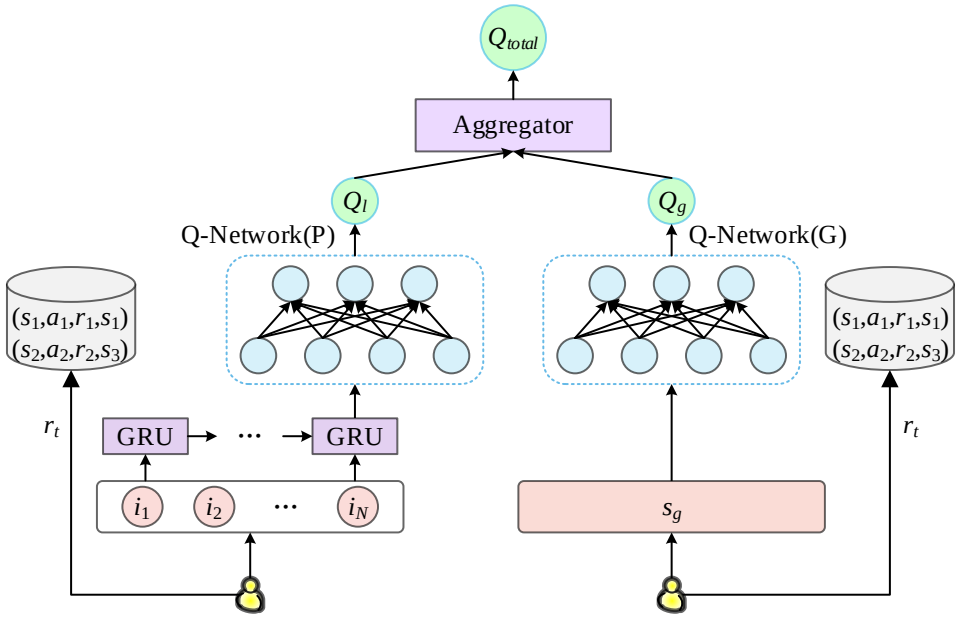


Figure 3: The recommendation system based on double DQN intelligent body

2.3.3 Model Training Algorithm

The proposed algorithmic framework in this chapter is based on the classical DQN structure, and the training process of the overall framework of the algorithm is given below:

(1) Initialize two intelligent body models using random weights; initialize the intelligent bodies' respective experience pools D_l and D_g.

(2) Randomly select a user, initialize the user's personal state as $\varnothing$, and obtain the user's contextual information to grasp the hot content of similar user groups.

(3) The two DQN intelligences calculate Q values according to their defined states.

(4) Calculate the total Q value and output the action according to the $\varepsilon - greedy$ policy.

(5) The local intelligent body collects user feedback and stores the interaction experience into the experience pool; the global intelligent body calculates whether the output action matches the current hotspot and stores the interaction experience into the experience pool.

(6) The two intelligences obtain small batches of data from their respective experience pools and optimize the value function network according to the training method of DQN.

(7) Return to step (2) until all users are trained.

3 Experimental results and analysis

3.1 Experimental design

3.1.1 Data sets

The number of students, questions, and interaction records included in each dataset are shown in Table 2, where Synthetic-5 is an artificial dataset and the rest are real datasets. ASSISTments2009 and 2015 are from the 2009 and 2015 data on the online education platform ASSISTments, respectively, and Statics2011 from the Engineering Statics course, Synthetic-5 is the dataset used for the DKT model, which simulates 4,000 virtual students answering 50 questions, and a homemade Civics dataset was also selected for the visualization of the experiment.

Table 2: Dataset introduction

Data set	Student number	Topic number	Interactive recording
ASSISTments2009	4152	110	325633
ASSISTments2015	19835	100	683815
Statics201	335	1225	189288
Synthetic-5	4000	50	20000

3.1.2 Contrasting models

In this paper, four representative knowledge tracing models were selected for comparison experiments, namely, DKT, DKVMN, CKT and SAKT, as follows:

DKT: The first use of recurrent neural networks to handle knowledge tracking tasks, and is a widely used classical model in the knowledge tracking field.

DKVMN: uses a dynamic key-value pair memory network to store and update students' knowledge states about each knowledge concept, enhancing the interpretability of the model.

CKT: enriches student characteristics by considering students' prior knowledge and learning rate when modeling their states.

SAKT: Uses the Transformer structure to handle knowledge tracking, which is free from the structural constraints of RNNs and does not suffer from the long sequence dependency problem.

3.1.3 Experimental setup

In data preprocessing, student data with less than 3 answer records are deleted, and student data with more than 380 answer records are truncated, and the portion with more than 380 answer records is treated as one new student data.

The parameters of the model in this paper are set as follows: the learning rate of the model training is 3E-3, epoch is 35, the dimension of the topic embedding in Eq. is 12, the size of convolution kernel in TCN is 5, and there are 12 residual blocks in total, and the dilation coefficient of dilated convolution is d=2n, which is doubled every other residual block, and the initial value of n is 0. The initial value of the Dropout ratio in the residual module is set to 0.06.

3.2 Results of the Knowledge Tracking Experiment

3.2.1 Experimental results and analysis

In this paper, the area under the curve (AUC) and accuracy (Acc) metrics are used to measure

the prediction effectiveness of the ATCKT model against the comparison models, and the experimental results of the different models on each dataset are shown in Table 3. Compared with other models, the AUC and Acc values of the F-TCKT model are the highest on the four datasets, especially on the ASSISTments2015 dataset, where the AUC value is improved by 19.37 percentage points to 92.24% compared to the DKT.The Acc value is improved by 10.97 percentage points to 86.13% compared to the DKT.

Table 3: Comparison of experimental results of different knowledge tracking models

Model	ASSISTments2009		ASSISTments2015		Statics2011		Synthetic-5	
	AUC	Acc	AUC	Acc	AUC	Acc	AUC	Acc
DKT	82.11	77.34	72.87	75.16	81.65	80.92	81.36	74.29
DKVMN	81.52	76.61	72.72	74.99	81.65	81.17	82.89	75.56
CKT	81.73	77.12	72.07	74.86	82.96	81.44	82.55	75.35
SAKT	82.09	76.99	85.45	78.54	82.34	81.25	83.88	76.85
F-TCKT	**84.96**	**85.61**	**92.24**	**86.13**	**83.75**	**81.59**	**87.57**	**80.02**

3.2.2 Topic Embedding Vector Similarity Visualization

The trained topic embedding vectors implicitly contain rich topic feature information such as topic difficulty and knowledge concepts involved in the topic, which helps the neural network to perform feature extraction. In order to verify the conclusion, this paper randomly selects seven topics on the self-constructed Civics dataset, which are topics No. 1, No. 4, No. 5, No. 7, No. 9, No. 11, and No. 15, and calculates the cosine similarity between their embedding vectors, and the knowledge concepts mainly involved in each topic are shown in Table 4, and the calculation results are shown in Fig. 4. The values inside the squares indicate the similarity of the corresponding topics, the larger the similarity the darker the color of the squares.

Questions No. 1 and No. 4 related to the knowledge point “volume” have a high similarity of 0.72, and questions No. 7 and No. 9 related to the knowledge point “linear equations” have a high similarity of 0.66, while the similarity between other questions is very low. The experimental results show that the embedding vectors of the trained topics contain the correlation information between different topics, which improves the interpretability of the model.

Table 4: Exercise problem numbers and corresponding knowledge concepts

Topic	Knowledge concept	Topic	Knowledge concept
1	The basic principle of marxism	9	Patriotism
4	Socialism theory	11	The state of law is governed comprehensively
5	Core values of socialism	15	Community of destiny
7	National spirit		

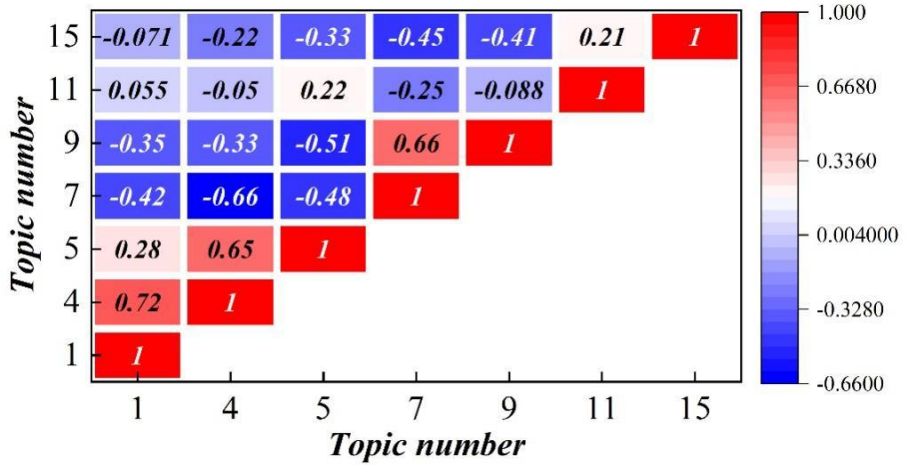


Figure 4: Cosine similarity of embeddings of exercise problems

3.2.3 Visualization of forecast results

In order to specifically illustrate the accuracy of the F-TCKT model predictions, this paper randomly selects a student in the self-constructed dataset and intercepts some of his/her outputs using the F-TCKT and DKT models respectively for visual comparison.The DKT model and the F-TCKT model prediction results are shown in Figs. 5 and 6, where the numbers 1, 4, 5, 7, and 9 on the left side of the picture indicate the question numbers, and the bottom of each column Two numbers indicate one interaction record, and the numbers between 0 and 1 in the table indicate the probability that the model predicts that the student will be able to answer the relevant question correctly the next time. In practice, if a student answers a question incorrectly consecutively, it means that the student's mastery of the question is poor and the probability of answering the question correctly in the future is small. When a student answered question No. 7 incorrectly consecutively, the F-TCKT model predicted the value of the probability that the student would answer question No. 7 correctly next time to be in the range of 0.2 to 0.5, whereas the predicted value of the DKT was in the range of 0.4 to 0.6, which is obviously not in line with the actual situation. This result shows that the Attention mechanism in the F-TCKT model can effectively identify the degree of contribution of the historical relevant answer records to the future knowledge state, which improves the interpretability and accuracy of the model.

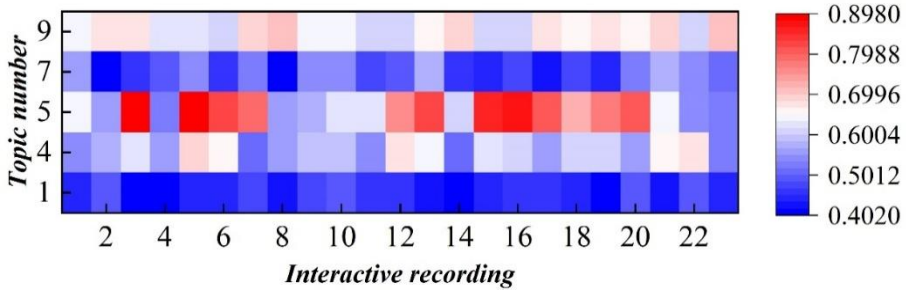


Figure 5: Visualization of prediction results of DKT model

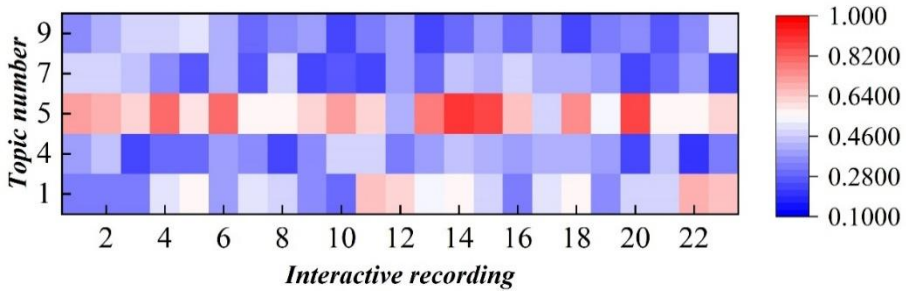


Figure 6: Visualization of prediction results of F-TCKT model

3.3 Deep reinforcement learning experiment results and analysis

3.3.1 Experimental procedure

(1) State representation. The knowledge tracking model serves as a student simulator for reinforcement learning, and the knowledge tracking model training dataset uses the Synthetic-5 dataset, so the state is set as a vector of length 50 in reinforcement learning. All elements in the vector are randomly initialized to simulate that different students have different levels of knowledge. After answering a question, the element in the corresponding position of the question is subjected to a plus one operation.

(2) Action selection. The action selection in the experiment uses all the 50 optional knowledge points. The output layer of the neural network is 50 nodes, and in action selection, an action vector of length 50 is output, and the position corresponding to the maximum value is selected as the action.

In the context of the application of recommended topics to students, two cases are considered: all actions are unique topics; all actions are knowledge points. In the former case, there cannot be multiple recommendations for topics that have already been recommended, and in the latter case, multiple recommendations can be made for topics of a certain knowledge point. In the training of deep knowledge tracking, combined with the application background, it is considered that the input of the model is an independent knowledge point, so in reinforcement learning, only the position of the element corresponding to the selected maximum value needs to be considered, without having to consider whether the knowledge point has been recommended or not.

(3) Environment Input and Output. Using the knowledge tracking model as a simulator for students to answer and assist in training the reinforcement learning network requires the design of the input vector for the knowledge tracking model. If we consider that each input is fixed as a vector of length 50, the meaning is that the student has gone through answering 50 questions. While the goal of reinforcement learning in this paper is to build a recommender system that tutors students' answers and rapidly improves their knowledge, so the input should be a vector whose length becomes longer with the training process. In the initialization, according to the initial state of the action selection, the action selection is M, at this time the input of the environment is {M}; the next moment, the recommended action is P, at this time the input of the environment is {M, P} This cycle stops when the action reaches a certain length.

3.3.2 Experimental results

In order to make the recommendation process can be more intuitively represented, this paper briefly describes the evaluation metrics used by the recommender system as the implied knowledge state mean cumulative value. In the experiment this cumulative value is the cumulative value of the immediate reward, and in the process of recommending M problems,

the experimental results are in the reward cumulative for M actions with the number of times of the training target.

Since the training target uses random values to simulate the initial state of different students during each initialization and the parameters are updated after a certain number of student trainings. So the cumulative reward value and the number of training times will form a variable curve. During this training process, the loss parameter should tend to decrease.

Figure 7 shows the change of loss function with the number of training steps, and the shaded part is the training error, the same below. In the process of recommending 3 topics at a time in sequence from 7 topics, due to the small number of recommended topics, the system achieves convergence in a short time and the cost parameter decays rapidly to 0. In the experiments, the optimization algorithm adopts the gradient descent Adam algorithm with the learning rate set to 0.01, the greedy probability parameter of the action is 0.8, the size of the memory pool is set to 1,000, and the size of the training batch is 64. In each training process the state vector is summed with a random variable in the range of (-0.3, 0.3).

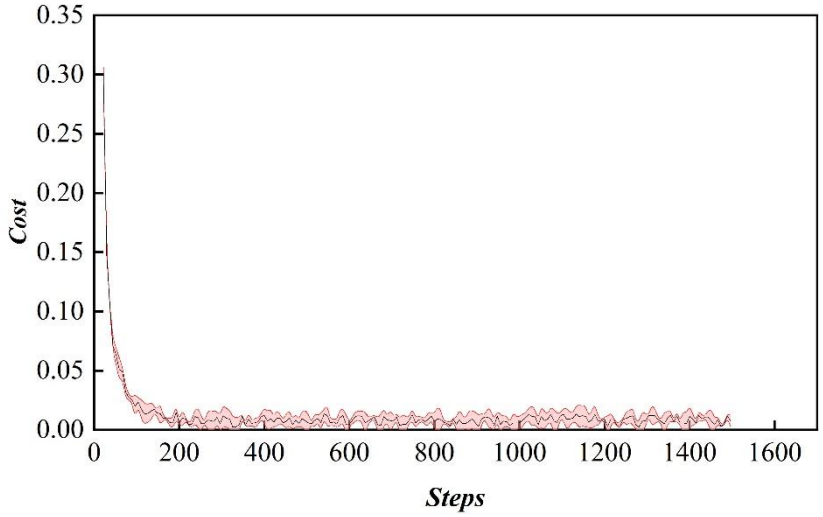


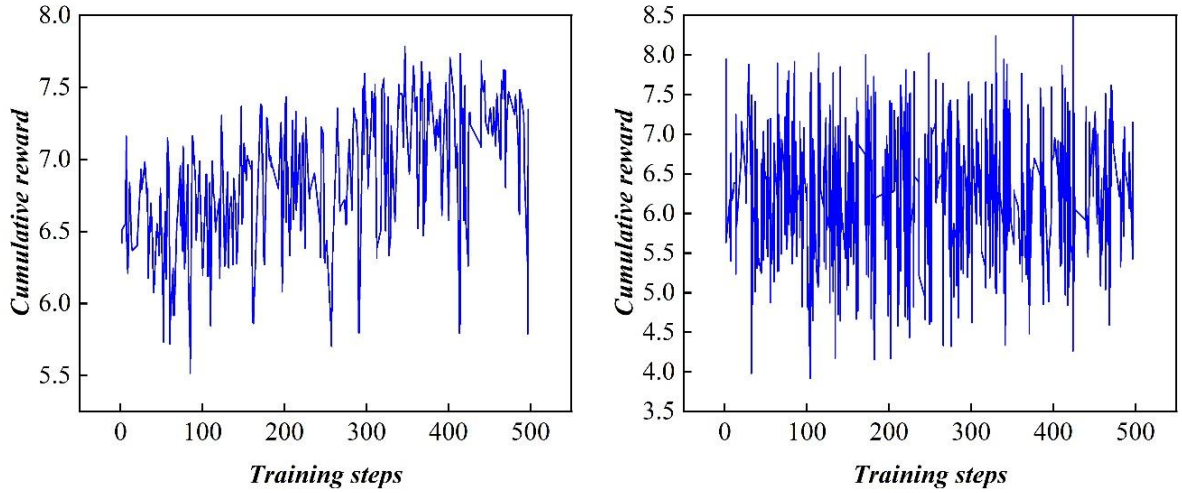
Figure 7: The loss function changes with the number of steps of the training

With a graph-based knowledge tracking model as a student simulator for reinforcement learning training.

Deep reinforcement learning uses the DDQN framework, and three sets of experiments are designed for different contents:

(1) Experiment 1, the presence or absence of a discount factor corresponds to the recommendation system using long-term boosting versus heuristic boosting, and comparative experiments are designed for these two recommendation methods.

Fig. 8 shows the comparison results of recommendation models with different discount factors. Figure 8(a) represents the change process of cumulative reward with the number of training steps when the discount factor is 0.4, and Figure 8(b) represents the change process of cumulative reward with the number of training steps when the discount factor is 0. It can be clearly seen that as the number of training rounds increases, the experiment with a discount factor of 0.4 gets better effect enhancement, which proves that the long-term recommendation effect is better than the heuristic recommendation result.



(a) γ=0.4 (b)γ=0

Figure 8: Comparison results of the recommended models for different discount factors

In order to represent the comparative results of the experiments more intuitively, the number of training steps is superimposed in accordance with the frequency of parameter updates of the target network, and Fig. 9 shows the variation of the average reward of the recommendation model with the number of parameter update rounds. The horizontal coordinate is the number of update rounds of the target network, and the vertical coordinate is the average value of rewards in that update cycle. It can be clearly observed that as the network is updated, the heuristic recommendation is similar to the long-term recommendation with discount factor in the early stage of training, and in the late stage of training, the long-term recommendation has been significantly more effective than the heuristic recommendation strategy.

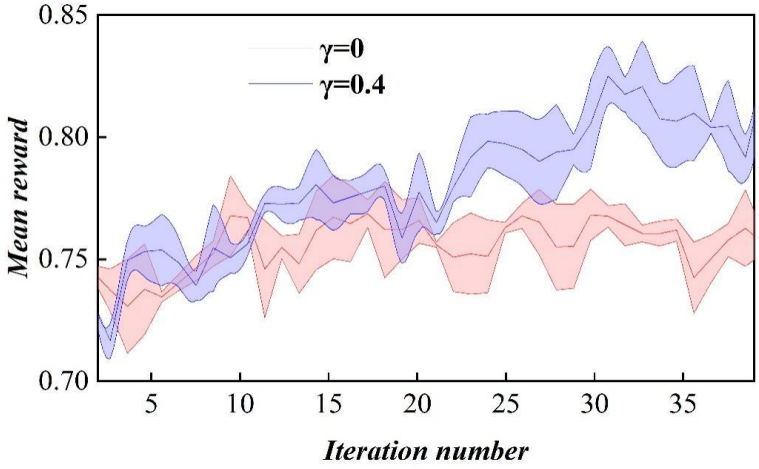


Figure 9: Average rewards vary with parameter update times

(2) Experiment 2, which verifies the necessity of intelligent recommendation, is conducted to compare the recommendation effect.

Fig. 10 and Fig. 11 show the comparison of the experiments with 10 and 5 questions extracted from the total number of 50 and 15 questions, respectively, and set up random action selection. It can be clearly observed that in both the recommendation models, the average reward obtained by the way of intelligent recommendation of topics compared to the reward of random action selection increases with the training process gap and has a clear advantage, and

finally stabilizes at a value that is significantly higher than the average reward of random action recommendation. The boost using intelligent recommendation is 0.08-0.1 higher than the average reward of random recommendation.

It can be assumed that the intelligent recommendation strategy for topics is effective, and through the training process, the topic scheme recommended to the learner can improve the learner's knowledge in a shorter period of time.

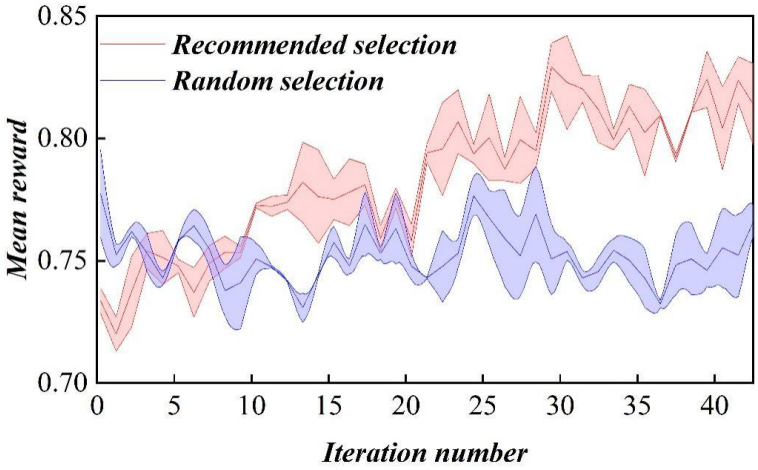


Figure 10: The recommendation of the 10 model is recommended

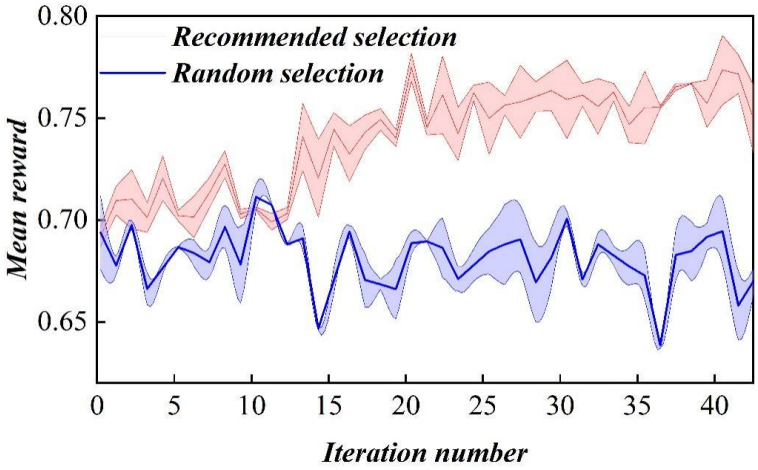
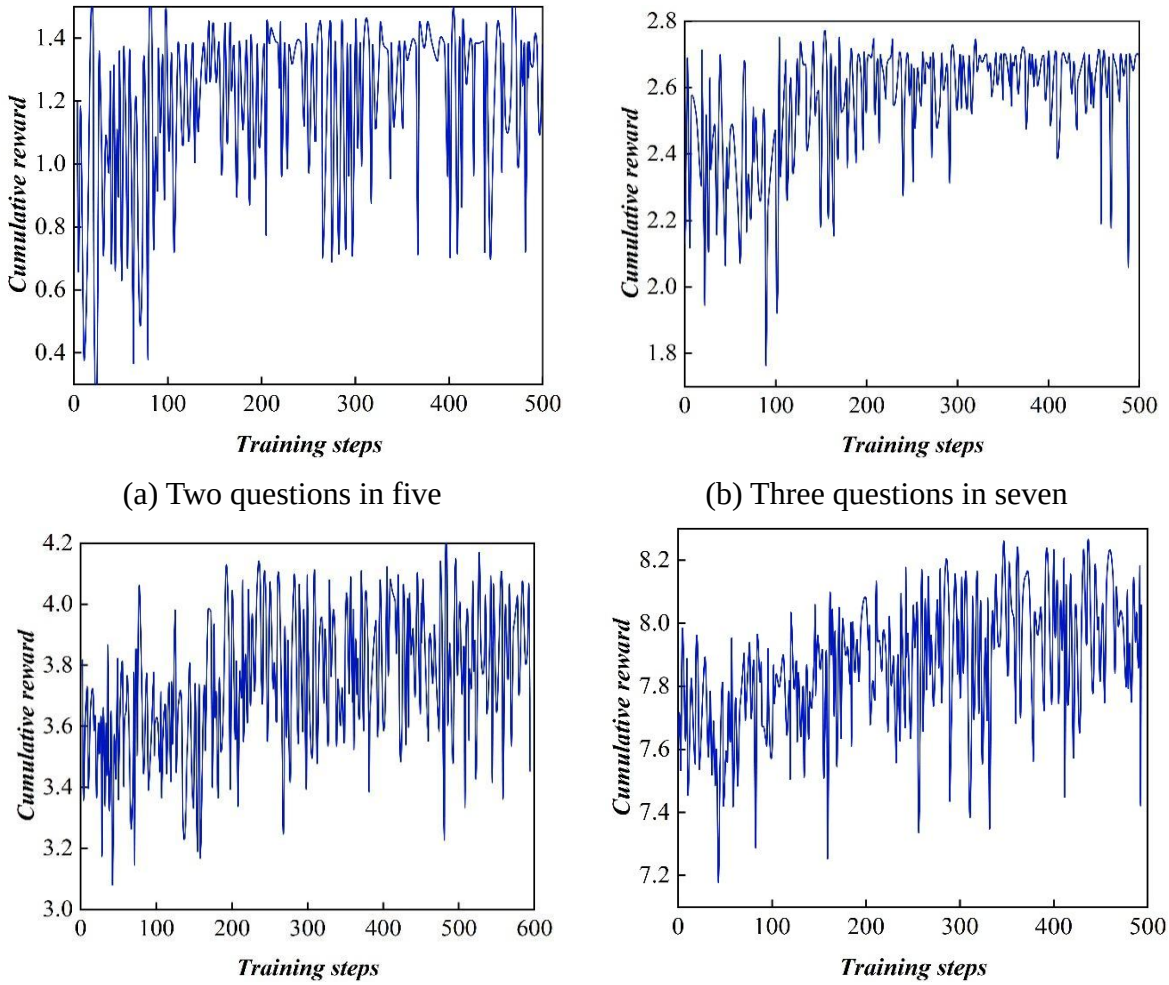


Figure 11: The recommendation of the 5 model is recommended in the topic

(3) Experiment 3, testing the recommendation ability of the recommendation model for different lengths of topic sequence recommendations.

Figure 12 shows the effect of using the deep reinforcement learning recommendation model in different application environments. When the total number of topics is small and the number of topics to be recommended sequentially is small, the model converges quickly and stably. As the total number of topics increases and the number of topics to be recommended increases, the convergence tends to be slow and the model does not converge accurately at the end of training. The reason for this phenomenon is that since the initial state of the student simulator is randomized each time, the recommended topic scheme will not be the same. This is also needed to personalize the recommendations for different students. It can be assumed that the model is applied to scenarios with a higher total number of topics.



(a) Two questions in five

(b) Three questions in seven

(c) Five questions in 15

(d) 10 questions in the 50 questions

Figure 12: The relationship between the recommended number and the reward curve

4 Conclusion

The significance of this project is to establish a precise Civics recommender system that assists students to rapidly improve their cognitive level in a network mimetic environment. A knowledge tracking model integrating forgetting factor and attention mechanism is proposed to model students' Civics level, and a double-Q deep reinforcement learning model is utilized to assist the synergistic development of accurate Civics.

The F-TCKT model improves the most on the ASSISTments2015 dataset, with AUC and Acc values improving by 19.37 and 10.97 percentage points compared to DKT. The visualization results found that when students answered Question No. 7 incorrectly consecutively, the probability value of students answering Question No. 7 correctly the next time predicted by the F-TCKT model was in the range of 0.2 to 0.5, and the Attention mechanism in the F-TCKT model could effectively identify the degree of contribution of the historically relevant answering records to the future state of knowledge, which improved the interpretability and accuracy of the model.

In the deep reinforcement learning part, doubleDQN algorithm is innovatively used to construct an accurate Civics recommendation system, and the result mainly proves that the recommendation effect gradually improves with the number of training times and tends to be

stable, which proves that the algorithm model can realize the rapid improvement of students' cognitive ability in a shorter time in the application of tutoring students. Future researchers can try to use other reinforcement learning algorithms to solve the problem of personalized recommendation.

About the Author

Wei Zhang (November 1984), female, Han nationality, from Weifang City, Shandong Province, Master's degree holder, lecturer at Sichuan Vocational College of Finance and Economics, research interests: Ideological and Political Education, Rural Revitalization.

References

[1] Shen, W., Yang, L., & Meng, C. (2025). Research on Ideological and Political Education of College Students in the Context of Douyin Network Ecology. Australian Journal of Electrical and Electronics Engineering, 1-13.

[2] Li, Q., & Ma, J. Q. (2020). China's research and prospects on discursive power of ideological and political education in internet environment. International Journal of Wireless and Mobile Computing, 18(3), 221-225.

[3] Wenqian, Y. (2021, June). Research and Analysis on the Construction of Network Ideology under We Media. In 2021 2nd International Conference on Artificial Intelligence and Education (ICAIE) (pp. 345-348). IEEE.

[4] Du, X. (2019, February). Research on the Ideological and Political Education Service Platform Based on the Campus Network Environment. In The International Conference on Cyber Security Intelligence and Analytics (pp. 181-187). Cham: Springer International Publishing.

[5] Wu, B. (2024). Research on the New Mode of Ideological and Political Education for College Students under the Network Environment. Curriculum Learning and Exploration, 2(1).

[6] Cheng, L., Varshney, K. R., & Liu, H. (2021). Socially responsible ai algorithms: Issues, purposes, and challenges. Journal of Artificial Intelligence Research, 71, 1137-1181.

[7] Al Ka'bi, A. (2023). Proposed artificial intelligence algorithm and deep learning techniques for development of higher education. International Journal of Intelligent Networks, 4, 68-73.

[8] Zhai, X., Chu, X., Chai, C. S., Jong, M. S. Y., Istenic, A., Spector, M., ... & Li, Y. (2021). A Review of Artificial Intelligence (AI) in Education from 2010 to 2020. Complexity, 2021(1), 8812542.

[9] Lin, Y. S., & Lai, Y. H. (2021). Analysis of AI precision education strategy for small private online courses. Frontiers in Psychology, 12, 749629.

[10] Zhang, T., Lu, X., Zhu, X., & Zhang, J. (2023). The contributions of AI in the

development of ideological and political perspectives in education. Heliyon, 9(3).

[11] Xu, C., & Wu, L. (2024). The Application of Artificial Intelligence Technology in Ideological and Political Education. International Journal of Advanced Computer Science & Applications, 15(1).

[12] Evans, A. L., Bulla, A. J., & Kieta, A. R. (2021). The precision teaching system: A synthesized definition, concept analysis, and process. Behavior Analysis in Practice, 14(3), 559-576.

[13] Yin, B., & Yuan, C. H. (2021). Precision teaching and learning performance in a blended learning environment. Frontiers in psychology, 12, 631125.

[14] Li, Y., & Mao, H. (2022). Study on machine learning applications in ideological and political education under the background of big data. Scientific Programming, 2022(1), 3317876.

[15] Zhang, W., & Zhang, D. (2025, June). An adaptive ideological and political teaching system using LSTM and optimization algorithms. In Second International Conference on Intelligent Transportation and Smart Cities (ICITSC 2025) (Vol. 13682, pp. 1022-1029). SPIE.

[16] Huang, Q. (2023). Research on Precise Ideological and Political Education Based on Improved K-means Algorithm for College Students' Portrait Construction. Journal of Artificial Intelligence Practice, 6(4), 58-64.