J. COMBIN. MATH. COMBIN. COMPUT. 127a (2025) 5747--5766

Journal of Combinatorial Mathematics and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Personalized Instructional Path Generation for Chinese Language Education Incorporating Reinforcement Learning

Zhengrong Liu[1,✉]

[1] *School of Humanities and Arts, Hunan International Economics University, Changsha, Hunan, 410000, China*

ABSTRACT

Personalized learning, in which learners set their own pace and select their own resources according to their own learning needs and characteristics, is the trend of Chinese education and teaching. In this paper, we design a personalized teaching path recommendation model for Chinese education based on reinforcement learning. The knowledge tracking prediction model LTKT is designed to integrate multiple knowledge points as information dimensions for model learning in the data preprocessing stage. The sparse self-attention mechanism is introduced into the encoder and decoder structure and embedded with location coding containing absolute and relative distances to enhance the model's perception of location information. Finally, the RL4ALPR algorithm is designed to model the changing knowledge level, the candidate learning item filtering algorithm is used to narrow down the scope of the recommended learning items, the reinforcement learning algorithm assumes the role of a recommender, and the degree of change in the knowledge level of the learner is regarded as a reward for the improvement of the reinforcement learning recommendation strategy. Simulation experiments are conducted on datasets such as ASSISTments and compared with baseline models such as KNN, GRU4Rec, Random, etc. The model in this paper has an F1 value and an AUC of 0.635 and 0.956 respectively in the evaluation of learning effect, which are the highest among the models. The study makes a useful exploration for the informatization of Chinese education and teaching.

Keywords: reinforcement learning, reward shaping, encoder, knowledge tracking, Chinese language education

1. Introduction

With the accelerated development of globalization, the importance of the Chinese language is becoming more and more prominent. As one of the world's largest countries in terms of population, China's influence is increasing day by day, and it has a pivotal position in the global arena. Against this background, Chinese language education is facing both opportunities and challenges. In terms of

✉ Corresponding author.
E-mail address: 18673167758@163.com (Z. Liu).
Received 10 January 2024; Revised 15 March 2024; Accepted 25 December 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-319

opportunities, more and more countries have begun to list Chinese as a second foreign language and set up Chinese language programs in schools, which provides a favorable environment for the expansion of China's economic, political and cultural influence [1-3]. The development of Chinese language education will further enhance the international status of Chinese language. By learning Chinese language, foreign talents can better integrate into Chinese society and culture and contribute to China's development, and Chinese talents can better go to the world and bring new impetus to the development of countries around the world. In terms of challenges, due to the relative shortage of Chinese language teachers, many regions are unable to provide high-quality Chinese language teaching, and the preparation of teaching materials is not close to the needs of the era of globalization; Chinese language education is not only about teaching language, but also covers Chinese culture and values, but due to the great differences in cultures of different countries and the current popular culture, which may easily lead to the conflict between the traditional culture and the contemporary values[4]. Nowadays, the demand for Chinese language learning is increasing.

Nowadays, the needs of learning Chinese are becoming more and more diversified, such as business communication, understanding of Chinese culture, and further education, which requires Chinese language education to provide personalized teaching programs and resources according to the needs of learners, in order to meet the needs of learning at different levels and for different purposes [5]. Personalized teaching is a current research hotspot, which is a kind of teaching method that tailors the teaching plan and teaching process according to students' characteristics and needs. It is through the understanding of students' learning characteristics, interests and learning difficulties, teachers can use different teaching methods and resources to meet students' learning needs, so that students can improve their learning at a pace that suits them [6]. Personalized teaching can also stimulate students' initiative and creativity in learning. Each student has different learning styles and interests, and teachers can adopt different teaching strategies for different students and encourage students to diversify their expressions and ways of thinking in order to improve students' learning motivation and cultivate their innovative and critical thinking skills [7-8].

Literature [9] designed a personalized teaching strategy for international Chinese language with the technical support of big data mining, which first classified learners into three categories: beginner, intermediate, and advanced, and then designed a targeted interactive teaching plan, set up challenging learning tasks, and provided corresponding teaching resources according to the characteristics of each category. Literature [10] analyzed the application of big data education platform and reading personalized teaching in Chinese education, and with the assistance of the platform, the effect of personalized reading was significantly improved and the quality was guaranteed. Literature [11] used fuzzy association rule mining and genetic algorithm to optimize Chinese personalized teaching, mainly through the two aspects of students' ability and preference to design a personalized curriculum. Literature [12] describes the problems in Chinese language teaching on the island such as poor language foundation and dialect impact, which lead to the challenges of Chinese personalized teaching, and puts forward targeted countermeasures to solve the problems. Literature [13] mentions that adaptive system is guided by teaching concepts such as personalized education, intelligent algorithms such as knowledge mapping do support to build personalized student models, provide personalized teaching for students, and give teaching strategies to apply it to Chinese language teaching to improve Chinese literacy with Chinese language practice and thinking practice, and build suitable learning paths and classroom plans according to their characteristics.

As a new hot research direction of personalized teaching in Chinese language education, current studies tend to focus on the implementation of personalized teaching strategies and the application of personalized Chinese language teaching with assistive technologies, lacking the introduction and planning of personalized teaching paths. Reinforcement learning, on the other hand, learns and optimizes a certain behavioral goal through trial and error, and is widely used in the fields of

personalized learning, adaptive learning, and self-directed learning in the field of education [14], which is a reliable and efficient approach for exploring and generating personalized teaching paths.

In this paper, we design a personalized teaching path recommendation model for Chinese education based on reinforcement learning. Firstly, we mathematically model the MDP settings in the recommendation scenario and the basic process of recommendation. Then a learning trajectory-oriented knowledge tracking prediction model (LTKT) is designed to solve multiple problems such as the lack of knowledge point information in the network, the distraction of attention to numerous less relevant test questions, and the neglect of the influence of learning ability, etc., which exist in the knowledge tracking domain using the Transformer architecture. Finally, the adaptive learning path recommendation model RL4ALPR is designed, which defines the learning links and cognitive structures suitable for learning path recommendation scenarios, gives a formal definition of adaptive learning path recommendation, and expresses the recommendation process through the analysis of real-life learning activities. Simulation experiments are conducted to verify the effectiveness of the algorithm.

2. Algorithm preparation

2.1. Overview of cognitive diagnostic models

The Cognitive Diagnostic Model (CDM) is an integral and fundamental component as a psychological assessment tool. In layman's terms, cognitive diagnosis is an assessment of cognitive levels aimed at discovering the state of a student's learning process, such as the student's mastery of knowledge points.

x_{ij} denotes the answer of student $i\left(i=1,2,\ldots,I\right)$ to exercise $j\left(j=1,2,\ldots,J\right)$. That is, when $x_{ij}=1$, it means that student i answered exercise j correctly, and conversely, when $x_{ij}=0$, it means that student i answered exercise j incorrectly. The values of θ and β are only 0 or 1, where β comes directly from the corresponding rows of the Q matrix, the matrix of correspondences between exercises and knowledge points labeled by the expert. $\beta_{jk}=1$ denotes the k th knowledge point that the student needs to master for answering the exercise j correctly, and vice versa; $\theta_i=\left\{\theta_{i1},\theta_{i2},\ldots,\theta_{iK}\right\}$ denotes the vector of knowledge points that the i th student possesses, where K denotes the number of all relevant knowledge points, and when the k st element in θ_i is 1, it means that the i nd student mastered the k rd knowledge point, and vice versa, it means that he/she did not master it. The potential response of the student i for Exercise j is $\eta_{ij}=\prod_{k=1}^{K}\theta_{ik}^{\beta_{jk}}$, $\eta_{ij}=1$ indicating that the student i has mastered all the knowledge points contained in Exercise j, and vice versa, indicating that the student i has not mastered at least one of the knowledge points contained in Exercise j. The probability of a student i answering Exercise j correctly is shown in Equation (1), provided that the student i's knowledge θ_{ij} is known:

$$P\left(r_{ij}=1\mid\theta_i,\beta_j\right)=g_j^{1-\eta_{ij}\left(1-s_j\right)^{\eta_{ij}}} \tag{1}$$

Each student and exercise feature is represented by a one-dimensional continuous variable, with θ denoting the student's ability and β denoting the difficulty of the exercise. The feature-to-feature modeling is done using a logistic regression form, the simplest version of which is Rasch, which describes only the relationship between student ability, the difficulty of the exercise and whether the student answers correctly; the GRM model additionally takes into account the differentiation of the exercises α_j. The IRT model is described as a generalized format where the probability of a student i answering an exercise j correctly is shown in Eq. (2):

$$P\left(r=1\mid\theta_i,\beta_j\right)=sigmoid\left(\alpha_j\left(\theta_i-\beta_j\right)\right) \tag{2}$$

Neural CDM, a neural network-based cognitive diagnostic framework, is a data-driven cognitive diagnostic model. Unlike DKT, given a student's answer record and Q -matrix, the goal of neurocognitive diagnosis is to tap into the student's knowledge point proficiency through a prediction process of whether the student answered the question correctly or not. In Neural CDM, each student has a knowledge point proficiency vector, which is obtained by multiplying the student's one-hot vector by a trainable matrix A. h^s The exercise knowledge point correlation vector Q_e comes directly from the given Q matrix.

2.2. Overview of Reinforcement Learning Algorithms

In this section, we will start with the basic concepts of reinforcement learning and describe the essential elements of a reinforcement learning problem. Introduce reinforcement learning related algorithms with the categorization criteria of online and offline learning, and outline the classical deep reinforcement learning algorithm Deep Q Network (DQN). Describe two techniques commonly used in reinforcement learning to mitigate the reward sparsity problem: reward shaping and course learning [15].

2.2.1. Intensive learning of basic concepts and solution methods. In reinforcement learning, there are two interacting objects: the intelligent body and the environment.

Intelligent body: can observe the state changes from the external environment as well as the rewards fed back from the environment, and learns the strategy by maximizing the accumulated rewards, and interacts with the environment by choosing the actions to perform according to the strategy.

Environment: the object for the intelligences to interact with. The state of the environment changes according to the actions performed by the intelligences, and accordingly, the environment feeds back rewards and punishments to the intelligences to guide them in their learning.

In order to estimate the expected payoff of strategy π, the state-value function and the state-action-value function are set up in the reinforcement learning base algorithm and are derived into an iterative form using Bellman's equation. Where the state-value function $V(s)$ represents the expectation of the payoff obtained by an intelligent body starting from state s according to the current policy π, $V^{\pi}(s)=\mathrm{E}_{a\sim\pi(a|s)}\mathrm{E}_{s'\sim p(s'|s,a)}\left[r\left(s,a,s'\right)+\gamma V^{\pi}\left(s'\right)\right]$. Similarly, behavior is introduced to form a state-action-value function $Q(s,a)$ representing the expected payoff that can be harvested by executing an action a for the current state s while following policy π, $Q^{\pi}(s,a)=\mathrm{E}_{s'\sim p(s'|s,a)}\left[r\left(s,a,s'\right)+\gamma Q^{\pi}\left(s',a'\right)\right]$. The value function can be viewed as an evaluation for policy π. An optimal strategy can be obtained by maximizing the state-action-value function, i.e., the optimal strategy $\pi^*(a\mid s)=\arg\max_a Q^*(s,a)$. according to Bellman's equation, the optimal state-action-value function $Q^*(s,a)$ can be computed by the value-iteration method as shown in Eq. (3):

$$Q^*(s,a)=\mathrm{E}_{s'\sim p(s'|\ s,a)}\left[r\left(s,a,s'\right)+\gamma\max_{a'}Q^*\left(s',a'\right)\right] \tag{3}$$

The temporal difference learning method is a classical approach in reinforcement learning for solving optimal policies for Markov decision processes. It combines dynamic programming and Monte Carlo methods by modeling a segment of trajectory in which the intelligent body evaluates the value of the state before taking action using Bellman's equation for each step, i.e., the value function.

2.2.2. Reward Shaping and Course Learning. The reward sparsity problem is a difficult issue that reinforcement learning intelligences often face in real-world tasks. Reinforcement learning intelligences learn through constant interaction with the environment and in a process of trial and error, with the goal of maximizing the accumulated rewards to obtain the optimal execution strategy [16]. Therefore, the rewards fed back to the intelligences by the environment are crucial for the learning of the intelligences.

The principle of reward shaping is to design some rewards artificially in a complex task to make the rewards dense. For example, in the task of training a robot to kick a soccer ball, only when the robot kicks the soccer ball into the goal will give a positive reward to the intelligent body, while in all other cases zero reward will be given. In this task, some stage rewards can be manually designed to guide the robot in the process of kicking the soccer ball into the goal.

2.3. Overview of Reinforcement Learning Based Recommendation Algorithms

In addition to the selection of recommendation algorithms, it is also very important to construct the recommendation system environment as an MDP. It can be said that the appropriateness of an MDP directly affects the learning efficiency and performance of the intelligences. In a recommender system environment, $MDP(\mathrm{S},\mathrm{A},\mathrm{P},\mathrm{R},\gamma)$ is usually set as:

State space S : Represented by the user's displayed or implicit features. For example, the user's history, the representation vector of the user's preference degree obtained through user modeling, etc.

Action space A : the set of items to be recommended. Action a performed at a certain moment is an item recommended by the recommendation algorithm to the corresponding user.

State transfer probability P : In the recommender system environment, the state transfer probability is invisible and it is derived through the interaction of the intelligence with the environment.

Reward Function R : It is determined by the specific task based on the user's feedback. It can be designed based on explicit outcomes such as click, skip, return, whether to buy or dwell time, or implicit outcomes such as user viscosity.

The recommendation flow of the recommendation system based on reinforcement learning is shown in Fig. 1. The environment includes four parts, user, user history log, user feature representation and reward calculation module.

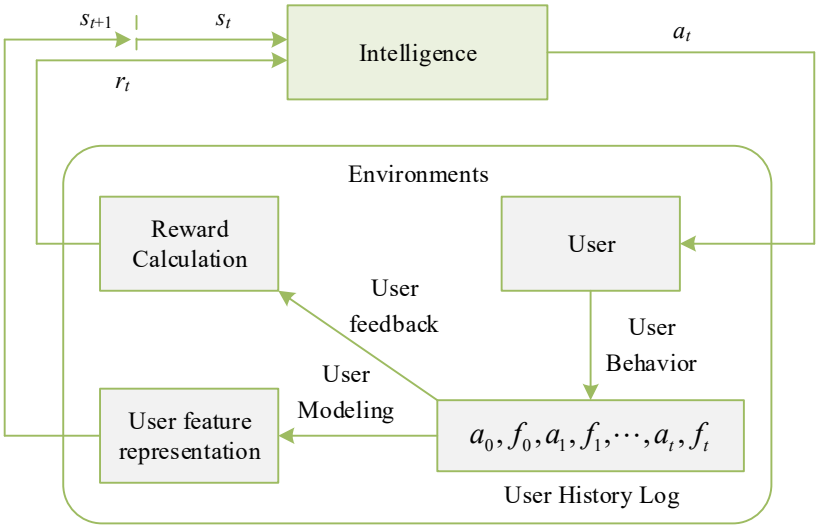


Fig. 1. Recommended process based on intensive learning

3. Design of personalized teaching path recommendation model for Chinese education

3.1. Input modules

The topic information is passed into the encoder to extract the correlation relationship between learning topics; the answer process information and labels are input into the decoder for dynamic

prediction; the topic information and answer results are passed into the learning ability extraction module to discover the learning ability state implied in the information.

Firstly, the knowledge points are mapped into high-dimensional knowledge vectors, and the knowledge vector table is constructed, and each knowledge point still has uniqueness after mapping, as shown in equation (4).

$$T=\left\{v_1,v_2,\cdots,v_j,\cdots,v_n\right\},v_j=Embedding\left(p_j\right) \tag{4}$$

From the topic perspective, the relationship between topics and knowledge points is one-to-many, and the number of knowledge points contained in each topic is different. According to the index of knowledge points corresponding to the topic, we find the corresponding knowledge vectors for summing operation, and fuse the information of multiple knowledge points in each vector dimension, and the fused knowledge vectors represent the original information of multiple knowledge points. Then the main feature extraction is performed to retain the main information of the high-dimensional knowledge vector, and the high-dimensional knowledge vector is downgraded to a single knowledge point value, but the distribution interval of the knowledge point value is small and the value is presented in the form of decimals. For this reason, a minimum-maximum normalization method is used to map the numerical intervals of the topic numbers to the knowledge point values, and the knowledge points are then represented by integers. The formula for the multi-knowledge point fusion processing of the topic is shown in equation (5).

$$count_i=PCA\left(v_{sum_i}\right),v_{sum_i}=\sum_{m=1}^{k}v_j \tag{5}$$

$$normcount_i=\frac{count_i-\min\left(Count\right)}{\max\left(Count\right)-\min\left(Count\right)}\left(b-a\right)+a \tag{6}$$

Where: is the number of knowledge points contained in the topic; $count_i$ is the value of knowledge points extracted from all knowledge vectors of q_i after summing operation and PCA main features; $Count$ is the set of all topics after processing $\left(count_i\in Count\right)$; a is the numerical lower limit of the topic number; b is the numerical upper limit of the topic number.

Multiple knowledge point information is fused into a new knowledge point in the data preprocessing stage. The knowledge point information is converted into embedding vectors using word embedding along with other input information and passed into the model.

3.1.1. Encoder and Decoder Modules. The encoder and decoder module consists of an encoder and a decoder. In the encoder, the input topic information is residually connected through the sparse self-attention mechanism and then subjected to summing and layer normalization to obtain the intrinsic associations between topics, which are then passed to the Conv1D network for feature extraction; such a structure is called a sparse self-attention layer. Multiple layers are often stacked in the encoder to map the topic information to a more abstract representation space so that the model can better capture the complex associations between topics. The outputs of the topic information obtained by the encoder are given as q_{out} and k_{out}. The attention layer is computed as:

$$O_{attn}=LN\left(input+SSA\left(input\right)\right) \tag{7}$$

$$O_{layer}=LN\left(O_{attn}+Conv1D\left(O_{attn}\right)\right) \tag{8}$$

Where: LN is the layer normalization computation; SSA is the sparse self-attention computation; and Conv1D is the one-dimensional convolutional feature extraction [17].

The decoder consists of a sparse self-attention mechanism and a traditional cross self-attention mechanism. The answer process information is obtained as output v_{out} by the sparse self-attention

mechanism, and the cross self-attention mechanism performs self-attention computation on q_{out}, k_{out}, and v_{out} using the traditional full self-attention mechanism to establish the connection between the question information and the answer process information.

To further improve the model's performance in modeling sequence information, this paper embeds position encoding containing absolute and relative distances in the sparse attention mechanism. The absolute position indicates the absolute positional relationship of each topic in the sequence, such as the first topic and the second topic in the sequence. The relative position indicates the positional difference between the topics in the sequence, such as the distance between the i th topic and the j th topic in the sequence, and the larger the relative distance is, the larger the interval between the two topics is.

Since only partially significant q and K matrices in the sparse self-attention mechanism are used for attention score computation, they do not satisfy the embedding condition in terms of both dimensionality and positional meaning. For this reason the sparse q_i and the corresponding indexes are found as shown in equation (9):

$$\{q_i, i\} = \max_{i,c}\left(\frac{q_i \overline{K}^T}{\sqrt{d}} - \frac{l}{L_Q}\sum_{i=1}^{L_Q}\frac{q_i \overline{K}^T}{\sqrt{d}}\right) \tag{9}$$

where $\overline{K} \in R^{c\times d}$ consists of c randomly selected k_j in K. The first item is the attention score of q_i and $\overline{K}$, and the second item is the mean of the attention score of Q and K, and the largest c q_i and the corresponding index value of Q are selected after subtraction.

The positional encoding of the corresponding position is intercepted based on the index value and summed with the sparse attentional scores to normalize the answer information for future moments into attentional weights, which are then updated V:

$$A(q_i, K, V) = soft\max\left(fillmask\left(\frac{q_i K^T}{\sqrt{d}} + pos_i\right)\right)V \tag{10}$$

where the function fillmask fills the upper triangle with a very small value close to 0, ensuring that the model focuses only on information from the current and previous moments, as well as information unknown about the future.

3.1.2. Learning Capability Extraction Module and Prediction Module. The question information and the information generated by answering the questions are mapped as high-dimensional vectors and then used as the input sequences of LAEM, and the learning ability features are extracted by 3-layer Conv1D convolution, and the convolution kernel of each layer halves the number of output channels, and the dimensions of the learning ability features are compressed, and the dimensionality of the compressed vectors is 1/8 of the original one.

The information of each input sequence consists of the learning trajectory of the same student, and the learning ability features obtained from the convolutional layer are averaged, and each sequence corresponds to get the learning ability features of the current student. After the dimensional alignment of the fully connected layer, the captured student ability features are passed to the prediction module together with the output of the decoder, and the bilinear layer is used to perform the feature extraction of the interactive fusion, thus realizing a more accurate prediction.

3.2. The RL4ALPR model

This section gives the general framework of the RL4ALPR model. Assuming that the length of the learning path is N, i.e., the recommendation is divided into a total of N time steps, next, the model details are explained in terms of t time steps.

RL4ALPR is divided into the following four sub-modules: knowledge level modeling (SAKT), candidate learning item screening (CN), reinforcement learning recommender (A2C), and reward computation. At each time step, SAKT explores the learner's potential knowledge level based on his/her historical interactions; CN filters a set of candidate learning items in the prerequisite graph based on the learning items answered by the learner in the previous time step, which serves as the action space for A2C: A2C provides the learner with the learning items to be answered; and rewards are passed to A2C for refining the recommendation strategy.

3.2.1. Knowledge level modeling. The structure of the SAKT model is shown in Figure 2. Firstly, each interaction tuple $x_i = \{e_i, score_i\}$ in X is operated $c_i = e_i + score_i \times E$ to transform the tuple into numerical form to obtain the input sequence $C = \{c_1, c_2, \ldots c_t\}$, where E is the number of exercises. The input sequence C is passed through the interaction embedding matrix $M \in R^{2E \times d}$, the position encoding matrix $P \in R^{n \times d}$ and the learning item embedding matrix $E \in R^{E \times d}$, respectively, to obtain the interaction embedding matrix $\hat{M} \in R^{n \times d}$ with position encoding and the learning item embedding matrix $\hat{E} \in R^{n \times d}$, where d is the hidden dimension, n is the maximum sequence length that the model can handle, and if the length of C is greater than n, C is divided into t / n subsequences, and if the length of C is less than n, iteratively on the left side of C fill the interaction x_i used for padding.

The embedding matrices $\hat{M}$, $\hat{E}$ are entered into the self-attention layer based on the deflationary dot product attention mechanism, which serves to assign a weight to each learning term of the previous answer, and the weights have an impact on the correctness of the model's predictions. The deflated dot product attention mechanism is shown in Equation (11):

$$Attention(Q, K, V) = soft\max\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{11}$$

In this, the query (Query) and Key-Value Pairs (Key-Value Pairs) are obtained using equation (12):

$$Q = \hat{E}W^Q, K = \hat{E}W^K, V = \hat{M}W^V \tag{12}$$

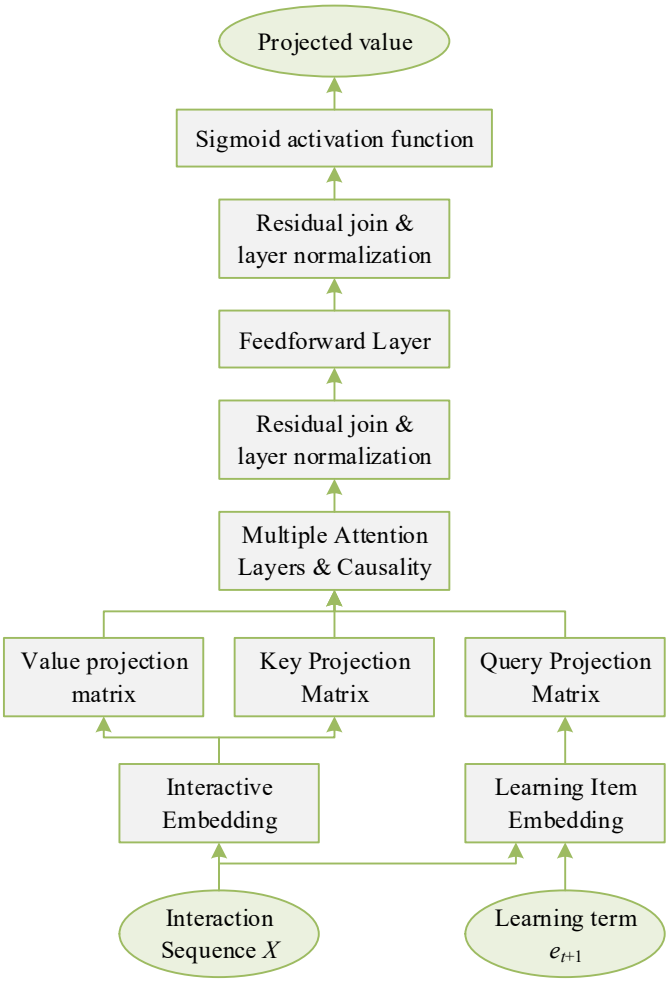


Fig. 2. Sakt model structure

W^Q, W^K, and $W^V \in R^{d\times d}$ are the query, key, and value projection matrices, respectively, which project the corresponding vectors to different spaces. In order to be able to process information representing different spaces at the same time, h linear projections are performed using the multi-head self-attention mechanism as shown in Eq. (13):

$$Multihead\left(\hat{M},\hat{E}\right)=Concat\left(head_1,head_2,\cdots head_h\right) \tag{13}$$

where $head=attention\left(\hat{E}W_i^Q,\hat{M}W_i^K,\hat{M}W_i^V\right)$, $W^o\in R^{hd\times d}$.

When predicting the learner's knowledge acquisition level at moment $(t+1)$, only the first t interactions should be considered, i.e., when $j>i$, for query Q_i, key K_i should not be taken into account. Therefore, a causality layer (Causality) is used to mask the weights of keys that are not considered.

The $S=Multihead\left(\hat{M},\hat{E}\right)$ obtained from the Multiple Self-Attention Layer is a linear combination of values, and in order to incorporate nonlinearity in the model and to take into account the interactions between the different latent dimensions, a feed-forward network is used to obtain the matrix F as shown in Eq. (14):

$$F=FFN(S)=\mathrm{Re}LU\left(SW^{(1)}+b^{(1)}\right)W^{(2)}+b^{(2)} \tag{14}$$

Where $W^{(1)}$, $W^{(2)}\in R^{d\times d}$, $b^{(1)}$, $b^{(2)}\in R^d$ are the parameters in training.

Residual connections are used to propagate lower layer features to higher layers. To prevent gradient explosion, gradient vanishing and to accelerate the convergence of the neural network, layer normalization is used in both the self-attention and feedforward layers.

Finally, the above matrix F is passed through a Sigmoid activation function to predict the probability of a student answering correctly to a learning item as shown in Equation (15):

$$\overline{SCORE} = Sigmoid\left(Fw + b\right) \tag{15}$$

$\overline{score_t}$ of each dimension value in $\overline{SCORE}$ denotes the probability that the learner answers e_i correctly, $i \in [2, 3 \ldots (t+1)]$, $Sigmoid(z) = 1/\left(1+e^{-z}\right)$, w are the weight matrices, and b is the bias.

The model trains the parameters by minimizing the cross-entropy loss between the predicted value $\overline{score_t}$ and the actual answer $score_i$, as shown in Equation (16):

$$L = -\sum_i \left(score_i \log\left(\overline{score_t}\right) + \left(1 - score_i\right)\log\left(1 - score_t\right)\right) \tag{16}$$

3.2.2. Candidate Learning Item Screening. In reality, learning activities tend to obey the human cognitive level, starting from the acquired knowledge and spreading outward to the knowledge associated with the known knowledge. Therefore maintaining logical relationships between knowledge is essential for learning recommender systems. Due to the complexity of logical relationships between knowledge points, this paper designs a cognitive navigation algorithm based on a certain center node on the prerequisite graph for quickly filtering candidate nodes associated with the center node. The set of learning items represented by a candidate node is the action space of the recommender described in Section 3.2.3, on which the policy network in the recommender selects a learning item. The filtered set of candidate learning items can avoid the recommended learning items violating the logic of human cognition, and at the same time, it can reduce the search space of the policy function in the intelligent body, which serves to accelerate the convergence.

3.2.3. Recommender Modeling. The role of the reinforcement learning A2C algorithm is to select the next moment action based on the current state, i.e., to recommend the learning items that need to be answered at the next moment for the learner based on the mastery level of each knowledge point that the learner currently possesses. Reinforcement learning A2C algorithm mainly consists of two parts: strategy network $\pi(a \mid s;\theta)$ and value network $v(s;w)$, both networks are forward fully connected neural networks. Policy network $\pi(a \mid s;\theta)$ is an approximation of the policy function $\pi(a \mid s)$, which controls the intelligent body to choose the action a of interacting with the environment, i.e., recommending the learning item k_t, and using the Softmax activation function, outputs a vector, the value of each dimension of which represents the probability that the learning item will be selected, and then randomly selects the learning item k_t according to the probability by using the strategy $\varepsilon - greedy$. Value network $v(s:w)$ is an approximation of the state value function $V_\pi(s)$, and its function is to evaluate how good or bad the current state s is, outputting a scalar. In contrast to the action value function $q(s,a;w)$ in actor-critic, $v(s;w)$ it depends only on the state s and is independent of the action a and thus $v(s;w)$ easier to train.

After observing the m trajectories $\left\{\left(s_{t+i}, k_{t+i}, r_{t+i}, s_{t+1+i}\right)\right\}_{i=0}^{m-1}$ within from t to $(t+m-1)$ moments, Multi-step TD Target is calculated first, as shown in Eq. (17):

$$mTDT_t = \sum_{i=0}^{m-1} \gamma^i \cdot r_{t+i} + \gamma^m \cdot v\left(s_{t+m}; w\right) \tag{17}$$

where γ is the discount rate. TDError is calculated using equation (18):

$$\delta_t = v\left(s_t; w\right) - mTDT_t \tag{18}$$

The parameters θ of the policy network $\pi(k_t \mid s_t;\theta_t)$ are updated using the policy gradient algorithm, and the derivation of the policy network $\pi(k_t \mid s_t;\theta_t)$ is performed using Equation (19):

$$d_{\theta,t} = \frac{\partial \ln \pi(k_t \mid s_t;\theta_t)}{\partial \theta_t} \tag{19}$$

Update parameter θ using gradient ascent as shown in equation (20):

$$\theta_{t+1} = \theta_t - \beta \cdot \delta_t \cdot d_{\theta,t} \tag{20}$$

The parameters w of the value network $v(s:w)$ are updated using the TD algorithm with Multi-Step TD Target, which is derived for the value network $v(s_t:w)$, as shown in Equation (21):

$$d_{w,t} = \frac{\partial v(s_t;w)}{\partial w_t} \tag{21}$$

Use Equation (22) to update parameter w:

$$w_{t+1} = w_t - \alpha \cdot \delta_t \cdot d_{w,t} \tag{22}$$

where $\delta_t \cdot d_{w,t}$ is the gradient of the loss function, and the loss function is shown in Eq. (23):

$$Loss(w) = \frac{1}{m}\sum_{i=0}^{m-1}\left[v(s_t;w) - mTDT_t\right]^2 = \frac{1}{m}\sum_{i=0}^{m-1}\delta_t^2 \tag{23}$$

The Monte Carlo approximation calculation $g(k_t;\theta)$ is used in updating the policy network as shown in Equation (24):

$$\begin{aligned} g(k_t;\theta) \approx & \frac{\partial \ln \pi(k_t \mid s_t;\theta)}{\partial \theta} \\ & \cdot\left[\sum_{i=0}^{m-1}\gamma^i \cdot r_{t+i} + \gamma^m \cdot v(s_{t+m};w) - v(s_t;w)\right] \end{aligned} \tag{24}$$

$$\begin{aligned} -\delta_t &= \sum_{i=0}^{m-1}\gamma^i \cdot r_{t+i} + \gamma^m \cdot v(s_{t+m};w) - v(s_t;w) \\ &= mTDT_t - v(s_t;w) \end{aligned} \tag{25}$$

The rationale for choosing $v(s_t;w)$ as the baseline here is as follows: $v(s_t;w)$ can be understood as a prediction of the future cumulative payoff U_t based on the state s_t at the moment of t, i.e., a prediction of the magnitude of the improvement of the learner's knowledge level:

$$v(s_t;w) \approx \mathrm{E}\left[U_t \mid s_t\right] \tag{26}$$

It can be used to evaluate the strengths and weaknesses of state s_t at the moment of t. The better the s_t, the greater the value of $v(s_t;w)$. $mTDT_t = \left[\sum_{i=0}^{m-1}\gamma^i \cdot r_{t+i} + \gamma^m \cdot v(s_{t+m};w)\right]$ can be interpreted as a prediction of how much the learner's knowledge will improve based on states s_t and s_{t+1}:

$$\begin{aligned} mTDT_t &= \sum_{i=0}^{m-1}\gamma^i \cdot r_{t+i} + \gamma^m \cdot v(s_{t+m};w) \\ &\approx \mathrm{E}\left[U_t \mid s_t, s_{t+1}\right] \end{aligned} \tag{27}$$

3.2.4. Calculation of incentives. After the execution of each action in a reinforcement learning algorithm, the environment provides an immediate reward r_t to guide the generation of the next action. For the learning path recommendation problem, when the recommended learning item is appropriate for the current learner and is beneficial to increase his/her knowledge level, the reward

should encourage this behavior and vice versa should penalize this behavior. Therefore, the reward function is defined as the degree of change in the learner's mastery level of the target knowledge point after answering the recommended learning item [18]. At each time step in the recommendation process, the reward function is set as follows:

$$r_t = \sum\left(y_{t+1,k} - y_{t,k}\right) \tag{28}$$

where k denotes the index of the knowledge point included in learning objective T, and $y_{t,k}$ denotes t the learner's mastery of the knowledge point k at the moment.

4. Simulation experiments

4.1. Learner performance prediction test

The overall performance of the model in this paper is first compared with the benchmark model. The results of the evaluation metrics of the two datasets are shown in Table 1. It can be found that the model proposed in this paper achieves better performance than other benchmark models in AUC and F1 in all datasets. Among them, the AUC and F1 are 0.925 and 0.907 on the ASSISTments dataset and 0.893 and 0.914 on the Junyi dataset, respectively.

In addition, the model in this paper obtains a significant improvement in effectiveness, and the following two advantages of the model in this paper can be considered as the key to obtaining the performance improvement:(1) The model in this paper models the impact of learning in both the temporal and spatial dimensions. (2) The model in this paper can track the spatial dimension learning impacts propagated along different types of relationships. This demonstrates the importance of modeling both temporal and spatial dimensional learning impacts simultaneously and of considering multiple relationships between knowledge structures in the knowledge tracing task.

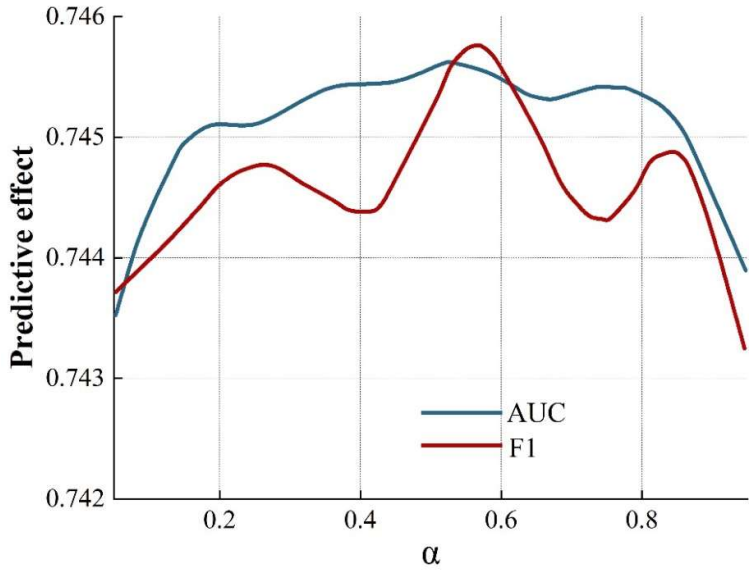
These phenomena suggest that considering knowledge transfer in the knowledge tracking process and modeling the process of propagating spatial dimension learning impacts in knowledge structures in a suitable way can significantly improve the predictive effectiveness of the model.

Table 1. Comparison of different knowledge tracking models

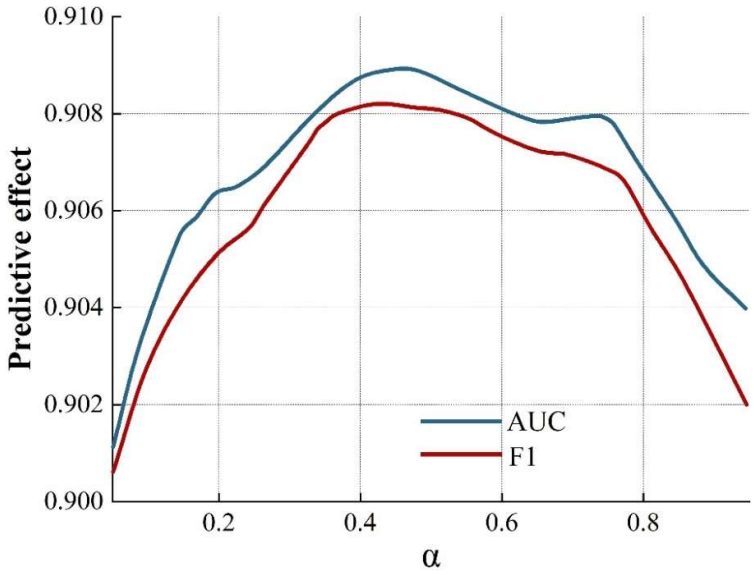
Data set	Eval	BKT	DKT	DKT+	DKVMN	GKT	This model
ASSISTments	AUC	0.805	0.684	0.612	0.643	0.742	0.925
	F1	0.697	0.794	0.721	0.604	0.682	0.907
Junyi	AUC	0.761	0.782	0.708	0.782	0.721	0.893
	F1	0.846	0.736	0.868	0.693	0.804	0.914

4.2. Parameter sensitivity tests

In this paper's model, the regulation parameter α plays a crucial role in regulating the weights of different learning influences propagated on conceptual similarity and learning sequential relationships in spatial dimensional influences. When α is small, the learning influence comes more from the learning sequential relationship. Conversely, when α is large, the model focuses more on learning influences propagated from conceptual similarity relations. Experiments were conducted for different α, where α was chosen from 0.05, 0.15..., 0.95. Figure 3 shows the impact on the model's effectiveness, (a) and (b) show the performance on the ASSISTments, Junyi datasets, respectively.



(a)ASSISTments



(b)Junyi

Fig. 3. Effect of model effect

When α increases, the recommendation effect of this paper's model increases at the very beginning. However, in both datasets, its prediction effect decreases after α is increased to a certain level. These results suggest that appropriately balancing the learning impact from learning sequential and conceptual similarity relationships is crucial to improve the model's prediction effectiveness.

4.3. Learning path recommendation application

4.3.1. Data sources. The learner data comes from 32 students who took the course “Chinese Literature” in the School of Computer Science and Technology of a university, of which 24 participated and filled out the valid fit questionnaire and 24 participated and filled out the valid measurement scale.

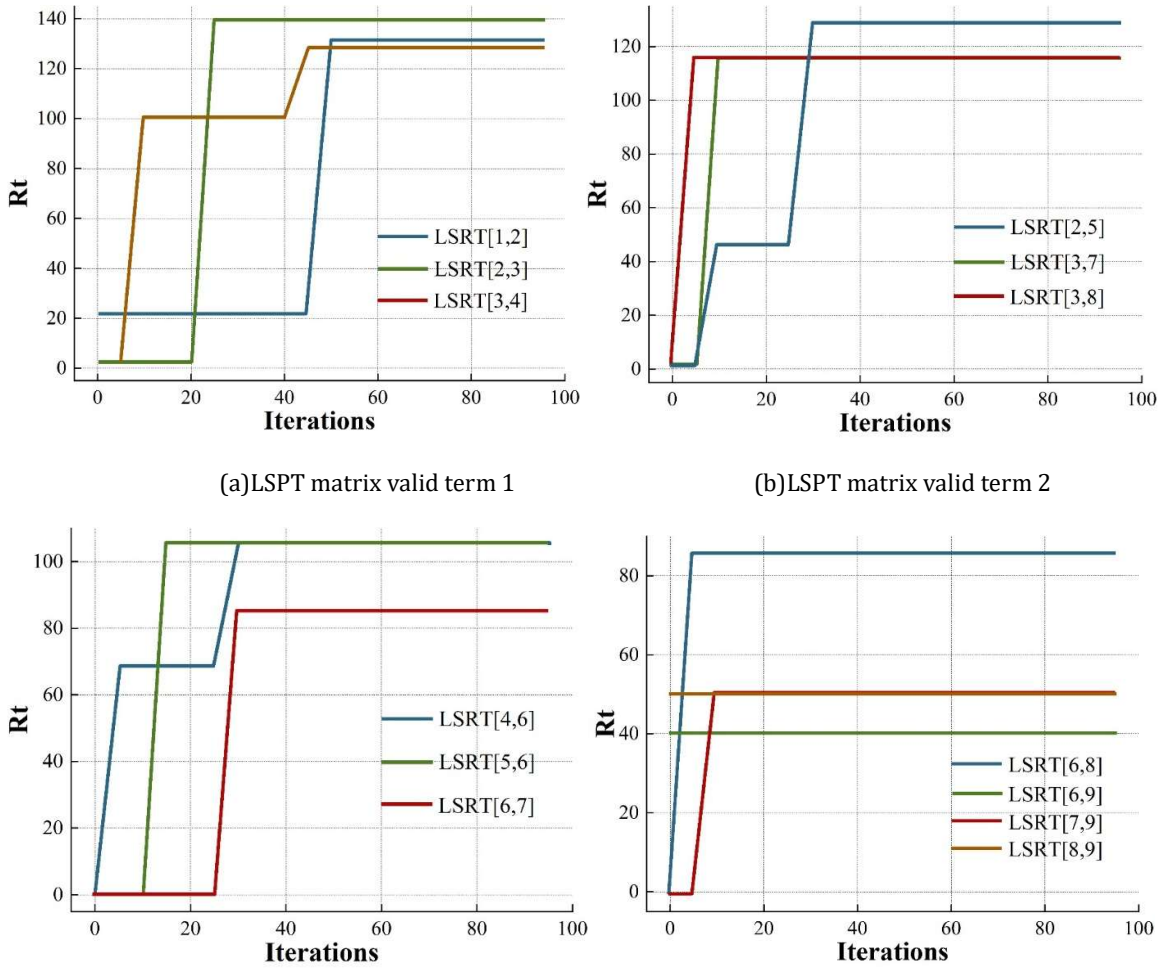
The learning resources data come from two units of virtual simulation experiments in the virtual simulation experimental teaching platform of a university majoring in Chinese language, including

eight virtual simulation experiments, including four principle demonstration experiments, three design interactive experiments, and one comprehensive experiment.

4.3.2. Experimental procedures. The manager needs to set the initialization parameters of reinforcement learning according to the scale of the problem, including the initialization iteration number N, the learning rate SS, the greedy coefficient Gr and the discount factor Dc, etc., of which the number of iterations is related to the number of the number of learning resources, and when the scale of the decision-making problem is large, a large number of iterative trainings are needed to complete it, and the decision-making problem in this paper is relatively simple, and the initial value of N can be set to 1000. The learning rate affects the learning speed of the intelligent body, when the learning rate is larger, the intelligent body is more easily affected by the past selection gain value, the intelligent body will quickly mature and generate results, on the contrary, the intelligent body will mature slower, here the initial value of SS can be set to 0.8. The greedy coefficient Gr affects the randomness of the results, when the greedy value is larger, the intelligent body will get the largest learning gain in the most greedy way, on the contrary, the intelligent body's selection will be more with a greedy way to get the largest learning gain. When the greedy value is larger, the intelligent body will get the maximum learning gain in the greediest way, and vice versa, the choice of the intelligent body will be more random, and the initial value of the greedy coefficient here is 0.9. The discount factor Dc is the weight of the gain in the memory and the gain of the current learning, and the larger the discount factor is, the smaller the weight of the gain in the memory and the larger the weight of the gain in the opposite way, and the initial value of the discount factor Dc here is 0.8.

Learners need to set two variables according to their own needs, i.e., the learning target value D and the fit weights. The learning target value can be set arbitrarily according to their own situation. When the learning target value is larger than the gain value of all learning resources, the learning process will stop after exceeding the maximum number of iterations, and the generated learning path recommendation queue will be the queue with the largest gain value under the current demand. When the learning target value is smaller than all the learning resources gain value, the learning process will stop when the learning target value is reached, and the generated learning path recommendation queue will be the largest gain value queue to meet the current demand.

When the four weights are set to (1, 1, 1, 1), the change of learning selection gain LSPT is observed, and the information can be derived as shown in Fig. 4. In Fig. 4, (a) to (d) are the different effective terms, i.e., the terms whose values change, in the LSPT matrix within 100 iterations, respectively. The other terms with unchanged values have no effect on the learning path and can be regarded as invalid terms. In Figure 4, we can see the change of Rt value during the iteration process, the various values of the learning selection gain table converge after 50 iterations, which indicates that under the current parameters, the learning gain reaches stability after 50 iterations, and the learning path is generated due to the size of the learning problem is not too large, and the convergence of each valid term is basically completed within 100 iterations, i.e., the intelligent body of the reinforcement learning has already obtained a stable solution, and the stable and the optimal learning path has been generated. From this process, we can see that the reinforcement learning intelligent body carries out human-like selection operation, and through hundreds of trial and error and continuous accumulation of experience, it finds the learning path that meets its own needs and maximizes the learning benefit. This part of the experiment proves the effectiveness of the adaptive learning path method based on reinforcement learning.



(a)LSPT matrix valid term 1 (b)LSPT matrix valid term 2

(c)LSPT matrix valid term 3 (d)LSPT matrix valid term 4

Fig. 4. Iteration times and Rt values

4.3.3. Analysis of results. The learning paths of 24 students with valid questionnaires are generated, and the results are shown in Table 2. From the table, it can be seen that after students choose different weights, the wisdom of enhanced learning will generate personalized learning paths based on students' individualized needs, combined with the fit information in the questionnaire, such as the table of student 1's learning gains weighted as 2, education weighted as 9, group weighted as 4, and technology weighted as 6, and the corresponding learning paths as 0→1→3→4→7→9, while student 2's learning gains weight is 8, education weight is 1, group weight is 1, technology weight is 9, and the corresponding learning path is 0→1→2→3→6→7. By comparing these two students, we can see that Student 1 is more concerned about the fit between learning resources and the learner himself, i.e., Student 1 cares more about the process of learning, whereas Student 2 pays more attention to the learning gain obtained from completing the learning, i.e., Student 2 cares more about the outcome of learning .

Table 2. Student choice values and corresponding generation learning path sequences

Student	FW	EW	SW	TW	Learning path
Student1	2	9	4	6	0-1-3-4-7-9
Student2	8	1	1	9	0-1-2-3-6-7
Student3	7	2	6	7	0-1-3-4-5-8

Student4	9	9	5	6	0-1-3-5-6-7
Student5	4	4	4	7	0-1-3-5-6-8
Student6	8	7	9	5	0-1-2-4-7-8
Student7	1	5	1	4	0-1-5-6-8-9
Student8	8	6	6	9	0-1-3-4-5-7
Student9	5	2	2	8	0-1-2-4-8-9
Student10	4	3	7	5	0-1-3-5-6-9
Student11	3	3	6	8	0-1-2-4-8-9
Student12	8	6	3	9	0-1-4-6-8-9
Student13	3	5	3	7	0-1-3-4-6-7
Student14	8	3	7	1	0-1-2-5-6-7
Student15	5	6	4	6	0-1-4-7-8-9
Student16	2	8	8	6	0-1-5-6-8-9
Student17	9	8	5	6	0-1-3-5-6-7
Student18	4	6	2	2	0-1-6-7-8-9
Student19	8	4	1	8	0-1-2-4-8-9
Student20	7	8	5	4	0-1-4-6-8-9
Student21	4	5	2	8	0-1-3-4-6-7
Student22	7	3	5	7	0-1-2-4-7-8
Student23	9	7	9	2	0-1-5-6-8-9
Student24	4	8	4	8	0-1-3-4-5-7

4.4. Comparison of Recommended Model Performance

A variety of mainstream recommendation models are selected as comparison models to compare the recommendation performance of learning path recommendation models within the learning task, and the specific experimental analysis and results are as follows:

4.4.1. Evaluation of learning outcomes. In this experiment, three recommendation models, KNN, GRU4Rec, and Random, are selected as the comparison models, and the learning task is “Chinese adverb usage”, and the length of the learning paths recommended by all models is fixed at 15 learning items. Among them, KNN is a user-based collaborative filtering recommendation model, which compares the cosine distance of the learning paths in the learning task, filters the similarity users as the neighbors of the current user from the user group, and recommends the next learning items for the new user; however, this model does not use the knowledge map and knowledge level tracking, so the learning task only includes the target knowledge points, but not the prerequisite knowledge points. GRU4Rec is a recursive neural network-based recommendation model that uses gated recurrent units (GRUs), where the input is the order of the learning paths in the learning task and the output is the probability distribution of the learning items that appear in the next step; however, this model also does not use knowledge mapping and knowledge level tracking, so the learning task includes only the target knowledge points and not the prerequisite knowledge points.Random is a stochastic recommendation model, where the recommended items are randomly selected from the learning

items of the learning task using a simple random sampling method; this model does not use knowledge level tracking although it uses knowledge mapping, so the learning task includes both the target knowledge points and the prerequisite knowledge points.

The results of the comparison of learning effect evaluation indexes of different recommendation models are shown in Table 3. In which IRT model and DKT model are used to calculate learners' initial mastery level and final mastery level of target knowledge points in the learning task respectively. Table 3 shows that this paper's model has the greatest improvement on learning effect, KNN and GRU4Rec models do not use knowledge graph and knowledge level tracking, only according to the learner's interest preference recommended learning program, its recommended learning path has almost no improvement on learning effect.Random model is better than KNN and GRU4Rec models due to the use of knowledge graph, so its recommended effect is better than KNN and GRU4Rec models, and its recommended effect is better than IRT and DKT models. GRU4Rec model, second only to the model model in this paper. The experimental results fully illustrate that the learning path recommendation is fundamentally different from the general path recommendation, and the knowledge graph, knowledge level tracking and recommendation algorithm all have a greater impact on the learning effect.

Table 3. Evaluation indicators of different recommendation models

Recommendation model	Precision	Recall	F1-Score	AUC
KNN	0.645	0.425	0.516	0.887
GRU4Rec	0.693	0.467	0.552	0.902
Random	0.748	0.481	0.571	0.529
This model	0.795	0.533	0.635	0.956

4.4.2. Evaluation of learning effectiveness. In this experiment, the Random model was selected as the comparison model, and the learning task was "adverb usage", and a virtual learner was randomly generated. The virtual learner's initial mastery level of "Adverb Usage" is 0.4885, and the learning goal is that the mastery level of "Adverb Usage" is greater than 0.9. The specific experimental procedure is as follows:

① Use the model in this paper to recommend a learning path for this learner, the path includes 8 learning items, and the mastery level of the target knowledge point reaches 0.9025.

② Use the Random model to randomly recommend multiple learning paths for this learner, and use the simple random sampling method to extract the learning items in each path.

③ The number of learning items in the learning path (i.e., the length of the learning path) is used as a quantitative evaluation index of learning efficiency, and the probability of the learning path recommendation is designed to facilitate the evaluation.

Event A represents the learning path with the number of learning items greater than, equal to, or less than 8, respectively, m represents the number of outcomes contained in A, and n represents the total number of possible outcomes.

The learning path recommendation probability of the Random model is shown in Fig. 5, which shows that: by conducting 1000, 5000 and 10000 random trials of the Random model, the P(A) value gradually stabilizes, and 4.65% of all the learning paths recommended by the Random model are better than the model of this paper, 39.71% are equal to the model of this paper, and 55.64% are inferior to the this paper's model. Among them, among the 499 learning paths that are better than this paper's model, all of them have a learning path length of 7. Among the 5564 learning paths that are inferior to this paper's model, there are 3693 learning paths that have a learning path length of 9, 1576 learning paths that have a learning path length of 10, 248 learning paths that have a learning path

length of 11, and 19 learning paths that have a learning path length of 12. Overall, 95.35% of the learning paths recommended by the model in this paper are better than or equal to the Random model.

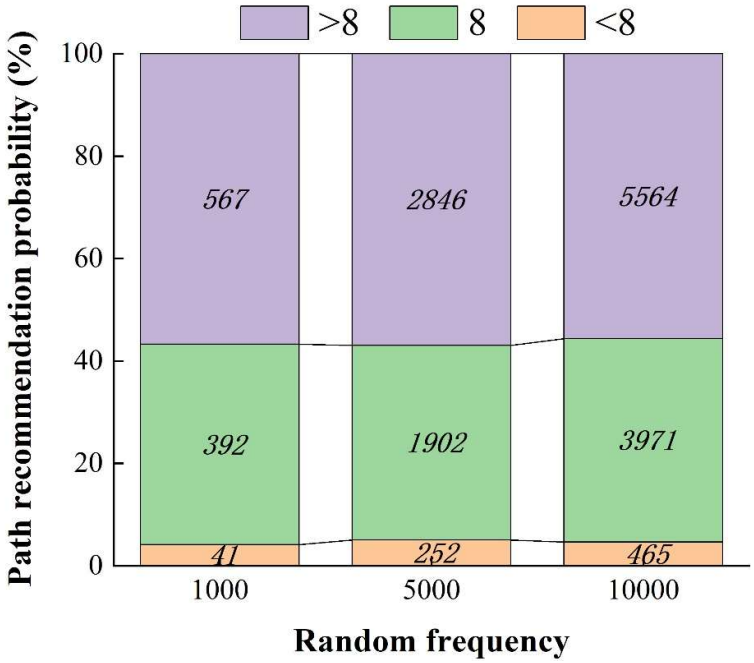


Fig. 5. Model learning path recommendation probability diagram

From the above analysis, it can be seen that the learning path recommended by the model in this paper is not 100% better than or equal to the Random model, the reason mainly lies in the model training, taking into account that the training time of the model should not be too long, so the algorithm convergence conditions are set. Experiments have proved that if the convergence conditions of the algorithm are improved, the performance of the recommendation will also be improved.

④ Using the Root Mean Square Error (RMSE) metric, the deviation value of the learning path length between the recommended learning paths of the two models and the optimal learning path is evaluated. In this experiment, the optimal learning path length is 7. The lower RMSE value represents the better performance of the model. The experimental results show that the RSME value of this paper's model with the optimal learning path is 1, and the RSME value of the Random model with the optimal learning path is 1.9385, which shows that the performance of this paper's model is significantly better than that of the Random model.

4.4.3. Recommended performance evaluation. The environment of this experiment is the same as the evaluation of learning effects: three recommendation models, KNN, GRU4Rec, and Random, are selected as comparison models, and the learning task selected is “adverb usage”, and the length of the learning path recommended by all models is fixed at 15 learning items. For a learning item, there are four possible outcomes after the recommendation: ① the model recommends the learning item to the learner and the learner has mastered the learning item; ② the model recommends the learning item to the learner but the learner has not mastered it; ③ the learner has mastered it but the model does not recommend it; ④ the learner has not mastered the learning item and the model does not recommend it. In this experiment, four commonly used recommendation algorithms evaluation indexes, i.e., check accuracy rate, recall rate, F1-Score and AUC, are selected to evaluate the recommendation performance of different models, and the results are shown in Table 4. Table 4 shows that compared with the other three models, this paper's model has the highest values of check-accuracy, recall, F1-Score, AUC, and its recommendation performance is optimal; the Random model has higher values of check-accuracy, recall, and F1-Score than the KNN model and the GRU4Rec model

due to the use of knowledge graph, but its AUC has the lowest value. Overall, the model in this paper has the best recommendation performance, followed by the GRU4Rec model, the KNN model, and the Random model in that order.

Table 4. Recommendations for recommendations of different models

Recommendation model	Precision	Recall	F1-Score	AUC
KNN	0.645	0.425	0.516	0.887
GRU4Rec	0.693	0.467	0.552	0.902
Random	0.748	0.481	0.571	0.529
LPRLT	0.795	0.533	0.635	0.956

5. Conclusion

In this paper, we designed a personalized teaching path recommendation model for Chinese education based on reinforcement learning, and tested the model on the dataset to check the improvement of the recommendation effect of the designed algorithm. The final conclusions are as follows:

1) In the learner performance prediction test, the AUC and F1 of this paper's model are 0.925 and 0.907 on the ASSISTments dataset, and 0.893 and 0.914 on the Junyi dataset, respectively, and the performances are the highest among all models.

2) In this paper's model, the adjustment parameter α regulates the weights of different learning influences propagated on the conceptual similarity and learning sequential relationships in the spatial dimension influences. In both datasets, the predictive effect decreases after α is increased to a certain level. These results suggest that appropriately balancing the learning influences from the learning sequential and conceptual similarity relations is crucial for improving the model's prediction effect.

3) This paper's model has the greatest improvement in learning effect, and the two models, KNN and GRU4Rec, do not use knowledge graph and knowledge level tracking, and only recommend learning programs based on learners' interest preferences, and their recommended learning paths have almost no improvement in learning effect.

In summary, it can be concluded that the model in this paper meets the design expectations and can be applied to the actual teaching of Chinese education.

Funding

This work was supported by research on interdisciplinary integration and development strategy of information literacy in basic education curriculum in digital teaching (No. XJKX24A041);

This work was supported by Hunan Province first-class undergraduate professional construction point (Xiang Jiao Tong [2020] No. 248 Item 416).

References

[1] Li, M. (2020). A systematic review of the research on Chinese character teaching and learning. Frontiers of Education in China, 15, 39-72.

[2] Sharma, B. K. (2018). Chinese as a global language: Negotiating ideologies and identities. Global Chinese, 4(1), 1-10.

[3] Khan, M. A., Zaki, S., & Memon, N. (2022). Chinese as a mandatory foreign language at a higher education institution in Pakistan. South Asia Research, 43(1), 49-67.

[4] Gong, Y., Lai, C., & Gao, X. (2020). The teaching and learning of Chinese as a second or foreign language:

The current situation and future directions. Frontiers of Education in China, 15(1), 1-13.

[5] Sun, X. (2020). The importance of the Chinese language in today's international business. Journal of Suvarnabhumi Institute of Technology (Humanities and Social Sciences), 6(1), 601-610.

[6] Reber, R., Canning, E. A., & Harackiewicz, J. M. (2018). Personalized education to increase interest. Current directions in psychological science, 27(6), 449-454.

[7] Rasheed, S., & Tanveer Kouser, D. K. R. (2024). Effect of Cooperative and Individualized Learning Strategies on Critical Thinking in Elementary Students. Remittances Review, 9(2), 4108-4133.

[8] Makhambetova, A., Zhiyenbayeva, N., & Ergesheva, E. (2021). Personalized learning strategy as a tool to improve academic performance and motivation of students. International Journal of Web-Based Learning and Teaching Technologies (IJWLTT), 16(6), 1-17.

[9] Yu, S., Zhang, Z., Kang, K., Zhu, L., & Jiang, X. (2024, May). Discussion on individualized teaching strategies of international Chinese education based on data mining. In International Conference on Artificial Intelligence for Society (pp. 574-583). Cham: Springer Nature Switzerland.

[10] Xiong, X. (2020, December). Research on Chinese Individualized Reading Teaching Based on" Big Data" Education Platform. In 2020 2nd International Conference on Information Technology and Computer Application (ITCA) (pp. 668-671). IEEE.

[11] Teng, F. (2024). Fuzzy Association Rule Mining for Personalized Chinese Language and Literature Teaching from Higher Education. International Journal of Computational Intelligence Systems, 17(1), 266.

[12] Song, C. (2023). Research on the Current Situation and Countermeasures of Personalized Teaching of Chinese Language Courses in Island Areas. Transactions on Comparative Education, 5(4), 13-18.

[13] Li, Y., & Liu, Z. (2024). Research on individualized Chinese teaching based on adaptive learning system. In SHS Web of Conferences (Vol. 187, p. 01028). EDP Sciences.

[14] 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.

[15] Melis Ilayda Bal ,Cem Iyigun ,Faruk Polat& Huseyin Aydin.(2024).Population-based exploration in reinforcement learning through repulsive reward shaping using eligibility traces. Annals of Operations Research(prepublish),1-33.

[16] Takumi Aotani,Taisuke Kobayashi & Kenji Sugimoto.(2021).Bottom-up multi-agent reinforcement learning by reward shaping for cooperative-competitive tasks. Applied Intelligence(7),1-19.

[17] Yinqian Sun,Feifei Zhao,Zhuoya Zhao & Yi Zeng.(2025).Multi-compartment neuron and population encoding powered spiking neural network for deep distributional reinforcement learning. Neural Networks106898-106898.

[18] Yuling Huang,Chujin Zhou,Lin Zhang & Xiaoping Lu.(2024).A Self-Rewarding Mechanism in Deep Reinforcement Learning for Trading Strategy Optimization.Mathematics(24),4020-4020.