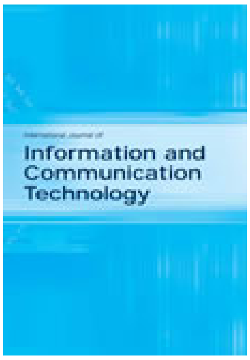


International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

Tracking domain knowledge in Chinese language education based on graph neural networks

Wenjuan Hu, Jing Fan, Yan Wang

DOI: 10.1504/IJICT.2025.10075271

Article History:

Received:	24 August 2025
Last revised:	12 November 2025
Accepted:	13 November 2025
Published online:	12 January 2026

Tracking domain knowledge in Chinese language education based on graph neural networks

Wenjuan Hu, Jing Fan* and Yan Wang

Hebei Institute of Mechanical and Electrical Technology,
Xingtai, 054000, China
Email: abc79599829@gmail.com
Email: fanjing@hbjd.edu.cn
Email: mjj84040225@163.com
*Corresponding author

Abstract: Knowledge tracking (KT) is a core task in the domain of integrated Chinese language education. However, traditional KT methods struggle to fully uncover the complex knowledge relationships within Chinese language education. To address this, this article designs a knowledge heterogeneous graph in the domain of Chinese language education, designs a heterogeneous graph neural network (GNN) to learn interactive relations among nodes, and extracts exercise node features as exercise representations. Then, a deep residual network is suggested to learn the interaction among exercise representations and students' answering abilities. Finally, a temporal convolutional network is used to track students' cognitive states and forecast the probability of them correctly answering the next exercise. Experimental results on the ASSIST and KDD datasets show that the proposed method improves prediction accuracy by at least 2.74% and 3.41%, respectively, enabling accurate forecasting of the mastery level of Chinese language knowledge points.

Keywords: Chinese language education integration field; knowledge tracking; graph neural network; GNN; deep residual network; temporal convolutional network; TCN.

Reference to this paper should be made as follows: Hu, W., Fan, J. and Wang, Y. (2025) 'Tracking domain knowledge in Chinese language education based on graph neural networks', *Int. J. Information and Communication Technology*, Vol. 26, No. 51, pp.34–49.

Biographical notes: Wenjuan Hu obtained her Master's in Mathematics from Hebei University of Technology in 2015. She is currently an Associate Professor at Hebei Institute of Mechanical and Electrical Technology. Her research interests include mathematical education, computer applications, and mathematical modelling.

Jing Fan received her Master's in Literature from Hebei University in 2007. She is currently an Associate Professor at Hebei Institute of Mechanical and Electrical Technology. Her research areas and directions include literature, language teaching, and traditional culture.

Yan Wang received her Master's degree from Hebei Normal University in 2015. She is currently a Professor at Hebei Institute of Mechanical and Electrical Technology. Her research areas and directions include literature, language teaching, and traditional culture.

1 Introduction

Chinese education, as an essential part of basic education, not only bears the responsibility of imparting knowledge of language and writing but also shoulders the mission of cultivating students' humanistic quality, thinking abilities, and aesthetic interests. However, Chinese knowledge is characterised by richness, complexity, and openness; it covers multiple dimensions such as vocabulary and grammar, and intricate relationships exist among the knowledge points (Shen et al., 2022). Traditional knowledge tracking (KT) approaches often concentrate on evaluating knowledge status from a individual dimension, making it difficult to comprehensively and accurately reflect the complex cognitive processes of students in Chinese learning (Zhao and Sun, 2024). As the deep learning technique emerging, KT methods in light of deep neural networks have increasingly been a research hotspot. Recurrent neural networks (RNN) and their variants are widely used in KT tasks, as they can process sequential data and capture time-dependent relationships during the learning process (Delianidi and Diamantaras, 2023). However, these methods still face challenges when handling knowledge with complex graph structures (Lai et al., 2021). In the Chinese knowledge system, the relationships between knowledge points resemble a graph structure. How to utilise deep learning algorithms to deeply explore the relationship features of knowledge points and achieve precise KT has significant research value.

Common KT models are generally divided into two categories. The first comprises traditional approaches, exemplified by Bayesian KT (BKT) (Liu et al., 2021a), and the other category consists of deep learning-based KT (DKT) methods (Piech et al., 2015). In the BKT model, the student's knowledge status is conceptualised as a set of binary variables, each denoting mastery or non-mastery of a detailed knowledge component, a hidden Markov model (HMM) (Alghamdi, 2016) is adopted to update the probabilities of each of these binary variables. This update mechanism captures the evolution of the student's knowledge state throughout the studying process. Lei et al. (2024) modified the BKT model and proposed a student-oriented method. This approach emphasises the uniqueness of each student, believing that each student should have a set of personalised parameters for all knowledge. Huber et al. (2024) introduced relationships between knowledge points, noting that knowledge points are not isolated, and changes in the mastery level of one skill are able to impact the learning status of related knowledge components. Takami et al. (2024) utilised BKT to customise a more suitable sequence of Chinese exercises for students to meet their personalised Chinese learning needs. Alotaibi and Papandreou (2022) effectively identified and extracted possible guessing components in students' answering processes by deeply analysing their multiple answering records and single answering records.

BKT-based models do not consider the impact of knowledge point difficulty on KT performance; they simply categorise exercises into certain knowledge points for prediction. BKT has relatively few training parameters, but the model's predictive accuracy lags significantly behind that of the DKT baseline. Compared to traditional KT models, DKT models learner proficiency within a high-dimensional continuous manifold, facilitating a more nuanced simulation of complex cognitive acquisition. Liu et al. (2019) integrated matrix factorisation technology into RNN, enabling the effective acquisition of exercise representations through matrix factorisation using only student interaction data, without including knowledge point information. Zou et al. (2020) input the textual information of exercises into a bidirectional LSTM to capture semantic information in the problems, and obtained tracing results through a fully connected network. The advantage of RNN-based models is their simple structure, but using a single vector to represent a student's knowledge status results in issues of poor interpretability and the inability to specify the student's grasp of individual knowledge points. Liu et al. (2021b) coped with this limitation through introducing a memory network; they proposed DKVMN. Unlike DKT, which stores knowledge states in a single hidden vector. In the DKVMN architecture, the key matrix defines the set of knowledge points, while the value matrix dynamically stores and updates the inferred mastery level of each student for the corresponding points. Xu et al. (2024) integrated student behavioural features, such as the number of attempts, with the student's learning ability based on DKVMN. Song et al. (2024) calculated the similarity among the current exercise to be forecasted and historical exercises the student has completed using cosine similarity when predicting student answering performance, and then aggregated the student's historical knowledge states weighted by the similarity.

Within KT, relational structures are frequently present. A contemporary approach to model these structures and more effectively tackle KT involves leveraging graph representation studying approaches, including GNN. Knowledge integration in language education exhibits characteristics such as complex interconnections, dynamic evolution, and multimodal fusion. Graph neural networks (GNNs) can effectively capture these features through structured relationship modelling, dynamic tracking, multimodal fusion, and personalised reasoning capabilities. Their graph structure inherently aligns with the networked nature of linguistic knowledge, while message passing and attention mechanisms support precise modelling of the knowledge integration process. Therefore, GNNs represent a highly promising technical direction for KT in language education. Techniques like GNN (Song et al., 2021) are widely used. Wu et al. (2022) suggested the GKT model in light of GNN. They framed the KT task as a node classification problem over time and addressed it with established graph learning techniques like message passing. Cui et al. (2024) proposed graph-based interactive knowledge tracing (GIKT), which utilised the relationship between exercises and knowledge concepts through graph representation to learn useful embeddings for answer prediction. Li et al. (2025) suggested structure-based KT, using the GNN structure to propagate feature information between knowledge concepts, considering the temporal characteristics of the sequence and the spatial features of the storage structure, and updating the learners' mastery status of knowledge concepts through a gating mechanism.

Researchers have proposed a series of KT models based on GNN, using GNN to model the intrinsic relations among knowledge concepts. These models consider the relations among exercises and knowledge points, but do not comprehensively consider the deep relationships among them. In addition, these methods usually measure the knowledge state of students from the overall learning cycle and ignore short-term fluctuations in knowledge states and students' answering abilities. To cope with the above issues, this article puts forward a Chinese education domain KT approach in light of GNN. The main work of this method is summarised into the following four aspects.

1 Collecting historical interaction data between students and knowledge in Chinese education integration, constructing a Chinese education domain knowledge heterogeneous graph with knowledge points, exercises, and other elements in the Chinese education integration domain, designing heterogeneous graph convolution to

learn exercise representations, taking into comprehensive account the difficulty level of the exercises, the coverage of knowledge points, the correspondence between exercises and knowledge points, and the similarity between exercises, thereby improving the comprehensiveness of exercise representations.

2 Using sliding window techniques to dynamically calculate student answering ability and combining it with exercise representations, inputting them into a deep residual cross network (DRCN). It can effectively differentiate changes in students' knowledge status and improve the understanding of the relationship between students and exercises through high-order feature interactions, thus enhancing prediction accuracy.

3 Based on the interaction of the above features, a temporal convolutional network (TCN) is used to track students' knowledge states. The TCN receives students' time-series response data and cognitive fusion feature information processed by the DRCN, extracts the knowledge state matrix of students at each time point and predicts the probability that students will answer the next exercise correctly.

4 Visualisation experiments and comparative studies were conducted on the ASSIST and KDD datasets. The outcome demonstrated that the proposed method achieved prediction accuracies of 92.81% and 95.09%, respectively, outperforming the comparison models. It demonstrated better adaptability to students' learning progress and enabled timely adjustments in assessing their mastery of integrated Chinese language education knowledge.

2 Relevant technologies

2.1 KT overview

KT dynamically tracks changes in students' knowledge state levels based on their answer records with a learning platform. Based on the obtained knowledge state levels of students, it can help teachers provide intelligent services to students. First, KT models enable learning platforms to offer personalised tutoring to students. Once a precise understanding of the students' knowledge status is achieved, the studying system can tailor more suitable studying plans for various students, thus enabling education to be adapted to the individual capabilities (Abdelrahman et al., 2023). Subsequently, students themselves can gain a clearer insight into their studying progress, helping them focus more on their learning obstacles and improving learning efficiency. During the learning process, the teaching system records student interaction data, including exercises, the knowledge concepts included in the exercises, and the students' responses.

A student's knowledge state in KT tasks is typically represented as discrete states, continuous states, skill graphs, vector spaces, probabilistic graphs, dynamic systems, or hybrid models. The specific choice of representation depends on task requirements, data availability, and model complexity. Modern KT models tend to favour continuous state or vector space models to more finely describe the gradual process of knowledge acquisition. KT tasks typically forecast the student's outcome on the subsequent problem in light of the estimated knowledge state, using the accuracy of predicting exercise performance to reflect the accuracy of evaluating the student's knowledge status (Liu et al., 2022). The KT task is generally defined as bellow. Given a student's record of answering interactions over T time $I = \{i_1, i_2, \ldots, i_T\}$, where i_j is the j^{th} answering record in the history. At time t, the answering record can be represented as a tuple, e.g., $i_t = <e_t, r_t>$, where e_t stands for the exercise answered at time t, and $r \in \{0, 1\}$ stands for the binary representation of the student's response result. The task of KT is to forecast the probability $\hat{r}_{t+1}$ that the student answers correctly in the subsequent $t + 1$ exercise e_{t+1} in light of the student's answering history sequence up to the time t. Its formal representation is as bellow.

$$\hat{r}_{t+1} = p\left(r_{t+1} = 1 \middle| i_1, i_2, \ldots, i_t, e_{t+1}\right) \tag{1}$$

2.2 Graph neural network

GNNs can learn node representation vectors that contain node feature information and contextual relationships, adopted to represent the state of knowledge points and students' learning history. The node embeddings learned by GNNs can be utilised to predict student proficiency across knowledge concepts and track their learning trajectories, and so on, providing support for personalised teaching (Ying et al., 2019). The chief idea of GNNs is to adopt a message passing scheme to aggregate node characteristic information in light of the connections between nodes (Khemani et al., 2024). In each layer of the computation process, each node updates its representation based on its own characteristics and the information from its neighbouring nodes. This information passing process can be iterated multiple times to gradually obtain a more comprehensive feature representation of the node. GNNs are divided into graph convolutional networks (GCN) and graph attention networks (GAT). The GCN model propagates information between nodes by applying a normalised Laplacian operator to aggregate features from adjacent nodes, and does not require additional parameter learning to determine the importance weights of neighbouring nodes during the computation process. This makes GCN highly computationally efficient in handling large-scale graph data, enabling fast forward and backward propagation processes. GAT introduces an attention mechanism that can automatically learn the significance weights of every neighbouring node for the central node. By calculating attention coefficients, GAT can dynamically adjust the aggregation weights in light of the characteristics of the neighbouring nodes and the central node, allowing the model to pay more attention to neighbouring nodes that have a greater impact on the central node.

GNN models take a graph as input. Based on the associations between nodes, they continuously receive and aggregate the representation information from neighbouring nodes, while also sharing their own representations with neighbours, ultimately achieving node embedding modelling. GNNs can not only model nodes, but also analyse and learn graphs at the edge level and the graph level. Taking GCN as an example, the model's input is a graph with C input channels, and through the implicit levels, it produces an output with F output channels. The learning process of GNNs relies on a local transition function that is shared by all nodes. Its definition is as follows, where x_v stands for the characteristics of node v, $x_{e[v]}$ stands for the characteristics of the edges connected to node v, $x_{e[v]}$ represents the state representations of nodes adjacent to node v, and $x_{n[v]}$ stands for the characteristics of the adjacent nodes.

$$h_v = f\left(x_v, x_{e[v]}, h_{n[v]}, x_{n[v]}\right) \tag{2}$$

3 Constructing a heterogeneous map of interdisciplinary knowledge in language education

To more comprehensively catch the complex relations among exercises and knowledge points in the integration of Chinese language education and domain knowledge, this paper constructs an exercise-knowledge heterogeneous graph to learn the representations of related problems. A heterogeneous graph can integrate various types of nodes and edges, enabling a comprehensive consideration of the difficulty characteristics of exercises and reflecting the similarity between exercises, thereby achieving more accurate and comprehensive problem representations. Heterogeneous graphs can incorporate nodes of various types, including knowledge points, questions, student characteristics, and practice scenarios. This approach breaks the limitations of traditional models that focus solely on knowledge points and answer results, enabling representations to encompass more comprehensive learning-related data. By learning the structural patterns of heterogeneous graphs, the model can better adapt to new knowledge points. This avoids over-reliance on existing data, leading to more stable performance in KT tasks across different scenarios and groups.

The input module is the data collection and pre-processing phase of the KT system, and its main task is to construct an information-rich interaction graph that can detail the interaction relations among students and knowledge points, as well as the structure within the students' learning social network. This module mainly consists of the following steps.

1 Data collection: First, the model collects historical interaction data $U = \{u_1, u_2, \ldots, u_n\}$ of students, where each student user u_i has a series of interaction records with knowledge points, including answer records, learning time, and discussions. At the same time, Chinese language education fusion domain knowledge point content data $N = \{n_1, n_2, \ldots, n_m\}$ is also collected, where each education i contains multimodal information text. Social network information $S = \{s_1, s_2, \ldots, s_k\}$ of users is also collected, recording the learning interaction behaviour of users. For the feature vectors of users and Chinese language education content, we use deep learning models respectively to extract.

2 Feature extraction: For knowledge point-related text t_{k_j}, the semantic vector $\vec{t}_{k_j}$ is extracted by natural language processing models. For students' social behaviour features $\vec{S}_i$, they can be obtained by analysing students' interaction data on the learning social network. The features of students and knowledge points are fused to form a unified representation. Suppose $\vec{l}_i$ is the feature vector of student l_i, and $\overrightarrow{k_j}$ is the fusion feature vector of knowledge point k_j, we can obtain the fusion features through the following equation.

$$\overrightarrow{l_i'} = f\left(\vec{l}_i, \vec{s}_i\right) \tag{3}$$

$$\overrightarrow{k_j'} = g\left(\overrightarrow{t_{k_j}}\right) \tag{4}$$

where f and g respectively represent the fusion functions of student features and knowledge point features. They may be some parameterised neural network models, such as fully connected layers or attention mechanisms.

3 Constructing the heterogeneous graph: This paper uses the PyTorch Geometric library to construct the heterogeneous graph, where the vertex features and construction of different types of edges are based on the following. First, construct vertex features. Chinese language education fusion domain-related exercise vertex features. Exercises have various levels of difficulty, and the vertex feature of each exercise is represented by its average correctness rate, calculated from the accumulated number of correct answers and the total amount of answers. Knowledge point vertex features: The features of the knowledge point vertices include two characteristics: the amount of covering exercises and the difficulty feature. The amount of covering exercises reflects the breadth of coverage of a knowledge point in exams or practices; the difficulty feature is calculated from the correctness rate of students on related exercises.

Then, construct edges of the heterogeneous graph.

- Exercise-knowledge point: Used to describe the correspondence among exercises and knowledge points, constructed based on the belonging relationship among exercises and knowledge points. If q_i involves knowledge point s_j, an edge is added between exercise q_i and s_j in the heterogeneous graph, indicating the correspondence among exercises and knowledge points.
- Exercise-exercise: Used to describe the explicit or implicit similarity relationships between different exercises. If exercise q_i and q_i include the same knowledge points, an edge is added between exercise q_i and q_j in the heterogeneous graph, indicating the similarity relationship between exercises.
- Knowledge point-knowledge point: Used to describe the explicit or implicit similarity relationships between different knowledge points. If exercise q_i involves both knowledge points s_i and s_j simultaneously, add an edge between s_i and s_j in the heterogeneous graph, indicating that knowledge points s_i and s_j have a similar relationship.

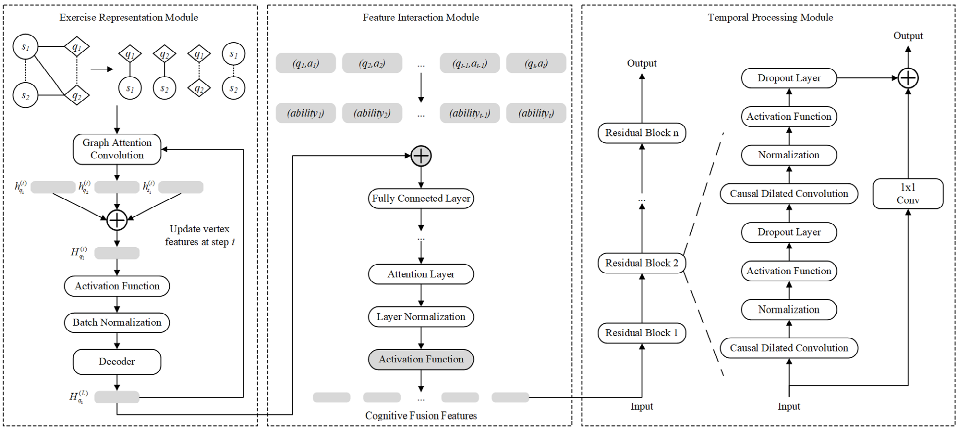
The input module ultimately converts the multimodal data in the Chinese language education fusion domain into structured data that can be processed by GNN, providing a foundation for subsequent knowledge state tracking and prediction.

4 A KT model for Chinese language education integration based on GNN'

4.1 Exercise representation learning based on GNN's

To address the difficulty of current research in fully exploring the rich and complex knowledge associations in Chinese language education and the dynamic characteristics of student learning processes, this paper suggests a Chinese language education integration domain KT model in light of GNN. The model consists of an exercise representation learning module, a feature interaction module, and a knowledge state extraction and forecasting module, as implied in Figure 1. The exercise representation module adopts a heterogeneous GNN to learn the interaction relationships between nodes and extract exercise node features as exercise representations. The feature interaction module uses DRCN to study the interaction between exercise representations and student answering abilities. The knowledge state extraction and prediction module uses a TCN to track student cognitive states and forecast the probability of students correctly answering the next exercise.

Figure 1 Structure of a KT model for language education integration based on GNN



The exercise representation learning module is influenced by the graph autoencoder (GAE) (Bai et al., 2024) architecture. This article suggests an approach in light of a heterogeneous GAE, which consists of an encoder and a decoder. The encoder part designs a heterogeneous graph convolutional network (HGCN) to process heterogeneous graph data containing multiple node and edge types. The HGCN draws on the core concept of the heterogeneous graph attention network (HAN) (Jia et al., 2023) to study node embeddings for the heterogeneous graph. Unlike HAN, which aggregates information through predefined meta-paths, HGCN uses graph attention convolution (GATConv) to dynamically adjust the weights of neighbour nodes, thereby learning embeddings for exercise nodes. For example, for the exercise node q_1, in the first layer of convolution, the initial feature $H_{q1}^{(0)}$ of q_1 is transformed linearly to generate intermediate features $z_{q1}^{(i)}$. Then, a weighted aggregation is performed on the adjacent nodes for all edge types $r \in R$. The weights $\alpha_{ij}^{(i,r)}$ are calculated by an attention scheme, which updates the feature of the exercise node to $H_{ql}^{(i+1)}$. After multiple layers of convolution, the final exercise representation $H_{ql}^{(L)}$ is generated. The formula is expressed as follows, where $\|$ denotes the concatenation operation, $N_r(q_1)$ stands for the set of adjacent nodes linked to the exercise node q_1 through edge type r, $BatchNorm(\cdot)$ is batch normalisation, and ReLU$(\cdot)$ is a nonlinear activation function.

$$z_{q1}^{(i)} = W^{(i)} H_{q1}^{(0)} \tag{5}$$

$$e_{ij}^{(i,r)} = \text{ReLU}\left(a_r^{(i)^T}\left(z_{ql}^{(i)} \middle\| z_j^{(i)}\right)\right) \tag{6}$$

$$\alpha_{ij}^{(i,r)} = \exp\left(e_{ij}^{(i,r)}\right) \Big/ \left(\sum_{k \in N_r(q_l)} \exp\left(e_{ik}^{(i,r)}\right)\right) \tag{7}$$

$$H_{q1}^{(i+1)} = \text{ReLU}\left(BatchNorm\left(\sum_{r \in R} \sum_{j \in N_r(q_l)} \alpha_{ij}^{(i,r)} z_j^{(i)} \right) \right) \tag{8}$$

$$H_{q1}^{(L)} = \sum_{r \in \mathbb{R}} \sum_{j \in N_r(q_l)} \alpha_{ij}^{(L,r)} z_j^{(L)} \tag{9}$$

The decoder is used to reconstruct edge information, aiming to catch the potential associations among exercises and knowledge points, as well as between exercises, thus generating high-quality exercise representations. The decoder maps node embeddings to a common space and calculates the edge reconstruction probability between each pair of nodes. For example, for exercise nodes q_1 and q_2, the decoder takes the node embedding vector $H_{q1}^{(L)}$ and maps it to the common space. The reconstruction probability R_{ij} is calculated as follows, where W_{dec} is the learnable linear transformation matrix in the decoder, and σ is the Sigmoid function.

$$z_{q_1} = W_{dec} H_{ql}^{(L)} \tag{10}$$

$$R_{ij} = \sigma\left(z_{q_1} \cdot z_{q_2} \right) \tag{11}$$

Through this architectural design, HGCN can effectively capture complex relationships in heterogeneous graphs by accurately reconstructing edge information to capture relationships between exercises, thus generating high-quality exercise representations.

4.2 Feature interaction based on deep residual cross networks

To deepen the model's understanding of the relationships between student-related exercises in the domain of Chinese education, this paper proposes a student response ability rating model by combining student answer accuracy and the number of attempts. A DRCN is designed to learn the complex relationships between student response abilities and exercise representations, generating student cognitive fusion features that integrate student learning ability and exercise representations.

To reveal the dynamic changes in students' learning states, this paper adopts a sliding window method to calculate students' exercise-answering accuracy across different time periods, aiming to capture students' short-term learning status in detail.

- Sliding window accuracy: The sliding window is an effective method for processing time series data. Since students' answering sequences have temporal characteristics, calculating students' exercise-answer accuracy during consecutive time periods can reveal their learning progress and changes in answering ability. The calculation of sliding window accuracy is as follows, where a_j represents the student's answering result at time point j, and W is the size of the sliding window.

$$Correct\ Rate_i = \left(\sum_{j=i+W-1}^{i} a_j \right) \Big/ W \tag{12}$$

- Student answering ability: Calculated by combining the sliding window answering accuracy and the amount of attempts. When a student has not encountered a particular knowledge point for a long time, they may face higher difficulty and more attempts, and the proposed scoring model will adjust the student's answering ability score accordingly to adapt to their actual learning state. The specific calculation is as follows, where $Correct\ Rate_i$ is the answering accuracy in the sliding window containing the current exercise i, and $attempt_i$ is the number of attempts for the current exercise. The weighting factor $(1 + 1/(attempt_i + 1))$ is used to adjust the score based on the number of attempts, ensuring a reasonable overall evaluation of the student's answering ability. The first '+1' is added to prevent calculation errors, such as division by zero. The second '+1' ensures that the answering ability score remains positive even if the student makes multiple attempts. This design provides a higher score for students who answer correctly on their first try, highlighting their ability to quickly master the knowledge points in the field of language education integration.

$$A_i = Correct\ Rate_i \times \left(1 + \frac{1}{attempt_i + 1}\right) \tag{13}$$

Designing high-order interactions between DRCN learning features before predicting the student's knowledge state effectively generates cognitive fusion features combining student answering ability and exercise representation. After concatenating the exercise representation and the student answering ability, the result is input into the DRCN, then goes through a fully connected layer for linear transformation and enters the attention layer. The attention layer uses a multi-head attention scheme to compute the relation weights among features, focusing on the most important features to enhance the model's processing of key information. In addition, residual connections are introduced in the DRCN. Residual connections can effectively avoid the gradient vanishing issue and improve the network's learning rate, accelerating the convergence speed. The equation is as follows.

$$X_{input} = Concat\left(Z_q, A\right) \tag{14}$$

$$X_{FC}^{(l)} = W_{FC}^{(l)} \cdot X^{(l-1)} + b_{FC}^{(l)} \tag{15}$$

$$Attention(Q, K, V) = \text{Softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \tag{16}$$

$$X^{(l)} = \text{ReLU}\left(LayerNorm\left(X^{(l-1)} + X_{Att}^{(l)}\right)\right) \tag{17}$$

where Z_q is the exercise representation, A is the student's answering ability at the current time, X_{input} is the initial input feature, $X_{FC}^{(l)}$ is the feature output after the fully linked level, and $X^{(l)}$ is the feature after processing through l layers, namely, the student's cognitive fusion feature.

4.3 Extraction and prediction of domain knowledge status in language education integration

This paper uses a TCN to extract the student's knowledge state. To solve the gradient vanishing problem, TCN introduces residual connections instead of simple connections between layers. The residual connection adds the input of each layer directly to the activation function before the output of that layer. TCN receives the time series data of student responses and the student's cognitive fusion feature information processed by DRCN, and extracts the student's knowledge state matrix Y at each moment from it. Its dimension is $R^{N \times K}$, in which N is the total amount of exercises answered by the student, and K is the dimension of the student's cognitive fusion features.

The student's knowledge state in the language education integration field at time t, y_t, is first mapped by a linear layer to generate a prediction vector Z_{t+1}. Then, Z_{t+1} is passed by a Sigmoid activation function layer to achieve the prediction probability p_{t+1}. The equation is as bellow, where W is the weight matrix of the linear layer and b is the bias term.

$$Z_{t+1} = WY_t + b \tag{18}$$

$$p_{t+1} = \sigma\left(Z_{t+1}\right) = \frac{1}{1 + e^{-Z_{t+1}}} \tag{19}$$

4.4 Model optimisation

Generally, during model training and optimisation, the model is trained by minimising a standard cross-entropy loss function, which quantifies the difference between the predictions and the ground-truth labels. Specifically, the model also incorporates a secondary training objective: to learn optimised exercise embeddings that encode information from the knowledge structure. This can avoid overfitting caused by using exercise embeddings alone. Therefore, the loss function is divided into two parts: one focused on maximising predictive accuracy and the other on learning semantically rich exercise embeddings.

$$L_1\left(\tilde{p}_{t+1}, p_{t+1}\right) = -\left(p_{t+1} \log \tilde{p}_{t+1} + \left(1 - p_{t+1}\right) \log\left(1 - \tilde{p}_{t+1}\right)\right) \tag{20}$$

In the second part, this paper designs the following method to calculate the loss for exercise representation.

$$L_2(i, j) = \begin{cases} 1 - \sigma\left(x^i \cdot (z^j)^T\right), & \text{if } r\left(e^i, z^j\right) = 1 \\ \sigma\left(x^i \cdot (z^j)^T\right), & \text{if } r\left(e^i, z^j\right) = 0 \end{cases} \tag{21}$$

It can be found that if exercise i contains knowledge concept j and their similarity is higher, then their dot product result will be larger, and $L_2(i, j)$ will be smaller. Ultimately, to enhance the trainable parameters throughout the entire model, this paper employs the following loss function to minimise the model's overall loss.

$$L = \frac{1}{L} \sum_{l=1}^{L} \frac{1}{T} \sum_{t=1}^{T} L_1\left(\tilde{p}_{t+1,l}, p_{t+1,l}\right) + \lambda \cdot \frac{1}{|E||K|} \sum_{i=1}^{|E|} \sum_{j=1}^{|K|} L_2(i, j) \tag{22}$$

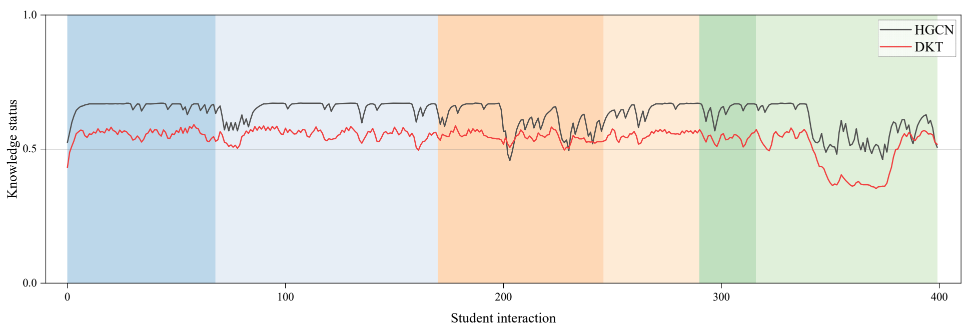
where L represents the amount of learners per batch during training, T stands for the amount of exercises completed by the learners, and λ is a hyperparameter used to adjust weights. It is worth noting that the learner index l is included in the total loss calculation formula because this model computes the loss for a batch of learners in each training process.

5 Experiment and performance analysis

5.1 Visualisation of Chinese education domain KT results

The experiments are conducted on Windows 11 with Intel Core i7-11370H CPU at 3.3 GHz. Python 3.9.10 is used as the programming language, and PyTorch 1.11.0 is used as the deep learning framework. Two public datasets, ASSIST (Xia et al., 2023) and KDD (Liu et al., 2023), are used in the experiments. The Chinese education domain knowledge is extracted from these datasets. After pre-processing, the ASSIST dataset is filtered to preserve records in which each knowledge point has been answered at least ten times. A total of 4,955 records are obtained, including responses from 237 students and 101 Chinese knowledge concepts. Similarly, this paper extracts records from the KDD dataset in which each knowledge point has been answered at least ten times. After pre-processing, 8,200 records are obtained, including responses from 168 students and 211 Chinese knowledge concepts. This paper classifies the dataset into training and test sets in an 8:2 ratio. The experiments are set with a maximum of 200 training epochs, a batch size of 64, and the hidden layer dimension set to 256.

Figure 2 Visualisation of changes in knowledge status, (a) knowledge state on the ASSIST dataset (b) knowledge state on the KDD dataset (see online version for colours)



(a)

(b)

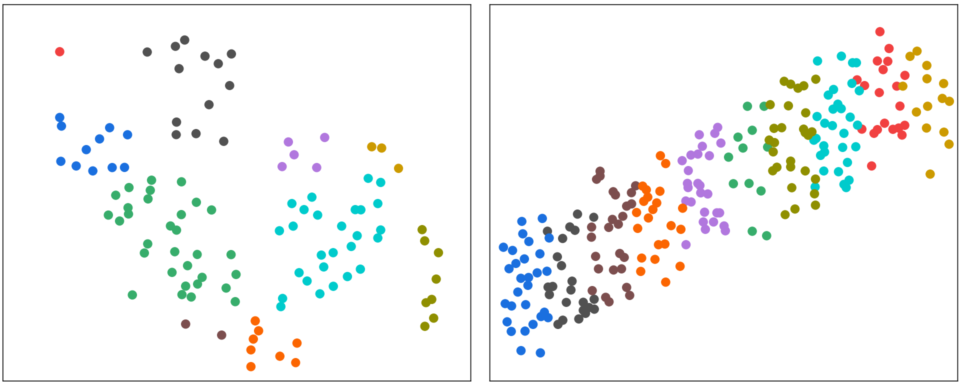
This paper visualises the knowledge state changes of a particular student from two datasets, as indicated in Figure 2. The x-axis represents the number of student interactions, which can be understood as interaction based on time changes, indicating the student's engagement in the studying process. The y-axis represents the measurement of the student's knowledge state, which is the result of summing and averaging the model's estimates of the student's mastery level for each exercise. The approach introduced in this work is named HGCN and is evaluated against the established DKT baseline. The background colour changes represent different days. This design helps observers understand the model's performance during different time periods. First, within the same day, i.e., in areas with the same background colour, HGCN shows more stable performance than DKT. This means that HGCN can maintain its accuracy in assessing a student's knowledge state at different time points throughout the day, without significant fluctuations. Second, when the background colour changes, HGCN can more precisely adjust its predictions to reflect knowledge retention or loss from one day to the next. This means HGCN not only captures the growth of a student's knowledge state during learning but also models the forgetting curve, offering a more comprehensive view of the student's learning process. Third, in the changes of the number of student interactions, especially changes within a single day, HGCN appears to be more sensitive and accurate in tracking subtle changes in the knowledge state. This indicates that HGCN can better adapt to students' learning rhythms and timely adjust the assessment of their mastery of knowledge.

5.2 KT performance comparison

To estimate the effectiveness of the GNN-based question representation learning module in HGCN for KT, 110 knowledge points in the dataset were clustered, with similar knowledge points represented using the same colour. For easier representation, the knowledge points were renumbered, and the outcome is indicated in Figure 3(a). In the figure, the knowledge points are divided into ten categories, with the red cluster representing knowledge points related to equations in mathematics. These shows that the GNN-based question representation learning module can effectively classify question knowledge points, acquire information about the knowledge points and the relationships between them, proving the effectiveness of this module in HGCN. Given the large number of questions in the dataset, this study selected some questions from ASSIST for clustering, with results shown in Figure 3(b). Different categories are represented by different colours. It can be seen from the figure that HGCN can effectively classify different questions.

In addition to conducting a qualitative analysis of KT for the proposed model, this paper also uses quantitative metrics such as accuracy (ACC) and AUC to objectively evaluate HGCN and benchmark methods LPKT (Zou et al., 2020), DKVMN-MRI (Xu et al., 2024), SAMKT (Song et al., 2024), SGKT (Wu et al., 2022), and DGEKT (Cui et al., 2024), and STHKT (Li et al., 2025). The quantitative metric evaluation results for different methods are shown in Table 1. On the ASSIST and KDD datasets, HGCN achieved ACC values of 92.81% and 95.09%, respectively, representing improvements of at least 2.74% and 3.41% compared to the baseline methods on both datasets. When comparing the AUC metric for KT prediction accuracy, HGCN achieved AUC values of 0.9638 and 0.9867 on the ASSIST and KDD datasets, respectively, representing improvements of at least 1.65% and 3.8% compared to the baseline method. HGCN can help models more effectively capture characteristics such as the difficulty of different questions when evaluating students' answers by designing heterogeneous GNNs. The designed residual cross-network can help models better identify the deep interaction between students' answering abilities and corresponding questions, avoiding the loss of feature information while alleviating the gradient disappearance problem in deep networks, thereby greatly improving KT effectiveness.

Figure 3 Exercise knowledge point classification results (see online version for colours)



(a) (b)

Table 1 Comparison of KT effectiveness between different methods

Method	*ASSIST dataset*		*KDD dataset*	
	ACC (%)	*AUC*	*ACC (%)*	*AUC*
LPKT	78.09	0.8062	80.03	0.7905
DKVMN-MRI	80.94	0.8417	81.92	0.8341
SAMKT	82.57	0.8592	84.64	0.8681
SGKT	85.21	0.8837	87.03	0.9004
DGEKT	86.93	0.8951	87.99	0.9079
STHKT	90.07	0.9482	92.08	0.9506
HGCN	92.81	0.9638	95.49	0.9867

6 Conclusions

In the field of Chinese language education integration, accurately tracking students' learning status and knowledge mastery is crucial for achieving personalised Chinese teaching and improving teaching quality. To address the issue where current research has not comprehensively considered the deep relationships among exercises and knowledge points, leading to low prediction accuracy, this paper proposes a KT method in the domain of Chinese language education integration based on GNN. Historical interaction data between students and integrated Chinese language education knowledge is collected. A heterogeneous graph of Chinese language education domain knowledge is constructed through knowledge points, exercises, and other elements in the domain of Chinese language education integration. HGCN is designed to learn the interaction relationships between nodes and extract exercise node features as question representations. Then, sliding window technology is used to dynamically calculate the student's answering ability, which is combined with the question representation and input into the DRCN. It can effectively distinguish changes in the student's knowledge status and improve the understanding of the relationship between students and questions through high-order feature interaction, thereby enhancing prediction accuracy. On the basis of the above feature interaction, the TCN is adopted to track the student's knowledge status. The TCN receives the time series data of student responses and the student's cognitive integration feature information processed by the DRCN, extracts the student's knowledge status matrix at each moment, and forecasts the probability of the student correctly answering the next question. Experimental outcome indicates that the suggested method has prediction accuracies of 92.81% and 95 on the ASSIST and KDD datasets, respectively.9%, showing a significant improvement over baseline methods, opening up new ideas for personalised teaching in Chinese language education.

Declarations

All authors declare that they have no conflicts of interest.

References

Abdelrahman, G., Wang, Q. and Nunes, B. (2023) 'Knowledge tracing: a survey', *ACM Computing Surveys*, Vol. 55, No. 11, pp.1–37.

Alghamdi, R. (2016) 'Hidden Markov models (HMMs) and security applications', *International Journal of Advanced Computer Science and Applications*, Vol. 7, No. 2, pp.17–32.

Alotaibi, O. and Papandreou, A. (2022) 'Bayesian nonparametric learning and knowledge transfer for object tracking under unknown time-varying conditions', *Frontiers in Signal Processing*, Vol. 2, pp.86–98.

Bai, L., Xu, Z., Cui, L., Li, M., Wang, Y. and Hancock, E. (2024) 'HC-GAE: the hierarchical cluster-based graph auto-encoder for graph representation learning', *Advances in Neural Information Processing Systems*, Vol. 37, pp.127968–127986.

Cui, C., Yao, Y., Zhang, C., Ma, H., Ma, Y., Ren, Z., Zhang, C. and Ko, J. (2024) 'DGEKT: a dual graph ensemble learning method for knowledge tracing', *ACM Transactions on Information Systems*, Vol. 42, No. 3, pp.1–24.

Delianidi, M. and Diamantaras, K. (2023) 'KT-Bi-GRU: student performance prediction with a bi-directional recurrent knowledge tracing neural network', *Journal of Educational Data Mining*, Vol. 15, No. 2, pp.1–21.

Huber, F., Bürkner, P-C., Göddeke, D. and Schulte, M. (2024) 'Knowledge-based modeling of simulation behavior for Bayesian optimization', *Computational Mechanics*, Vol. 74, No. 1, pp.151–168.

Jia, X., Jiang, M., Dong, Y., Zhu, F., Lin, H., Xin, Y. and Chen, H. (2023) 'Multimodal heterogeneous graph attention network', *Neural Computing and Applications*, Vol. 35, No. 4, pp.3357–3372.

Khemani, B., Patil, S., Kotecha, K. and Tanwar, S. (2024) 'A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions', *Journal of Big Data*, Vol. 11, No. 1, pp.18–25.

Lai, Z., Wang, L. and Ling, Q. (2021) 'Recurrent knowledge tracing machine based on the knowledge state of students', *Expert Systems*, Vol. 38, No. 8, pp.82–89.

Lei, T., Yan, Y. and Zhang, B. (2024) 'An improved Bayesian knowledge tracking model for intelligent teaching quality evaluation in digital media', *IEEE Access*, Vol. 12, pp.125223–125234.

Li, S., Shen, S., Su, Y., Sun, X., Lu, J., Mo, Q., Wu, Z. and Liu, Q. (2025) 'STHKT: spatiotemporal knowledge tracing with topological Hawkes process', *Expert Systems with Applications*, Vol. 259, pp.12–18.

Liu, F., Hu, X., Bu, C. and Yu, K. (2021a) 'Fuzzy Bayesian knowledge tracing', *IEEE Transactions on Fuzzy Systems*, Vol. 30, No. 7, pp.2412–2425.

Liu, S., Zou, R., Sun, J., Zhang, K., Jiang, L., Zhou, D. and Yang, J. (2021b) 'A hierarchical memory network for knowledge tracing', *Expert Systems with Applications*, Vol. 177, pp.49–55.

Liu, Q., Huang, Z., Yin, Y., Chen, E., Xiong, H., Su, Y. and Hu, G. (2019) 'EKT: exercise-aware knowledge tracing for student performance prediction', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 33, No. 1, pp.100–115.

Liu, Z., Liu, Q., Chen, J., Huang, S., Tang, J. and Luo, W. (2022) 'pyKT: a python library to benchmark deep learning based knowledge tracing models', *Advances in Neural Information Processing Systems*, Vol. 35, pp.18542–18555.

Liu, Z., Liu, Q., Guo, T., Chen, J., Huang, S., Zhao, X., Tang, J., Luo, W. and Weng, J. (2023) 'Xes3g5m: a knowledge tracing benchmark dataset with auxiliary information', *Advances in Neural Information Processing Systems*, Vol. 36, pp.32958–32970.

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L.J. and Sohl-Dickstein, J. (2015) 'Deep knowledge tracing', *Advances in Neural Information Processing Systems*, Vol. 28, pp.53–61.

Shen, S., Chen, E., Liu, Q., Huang, Z., Huang, W., Yin, Y., Su, Y. and Wang, S. (2022) 'Monitoring student progress for learning process-consistent knowledge tracing', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35, No. 8, pp.8213–8227.

Song, J., Wang, Y., Zhang, C. and Xie, K. (2024) 'Self-attention and forgetting fusion knowledge tracking algorithm', *Information Sciences*, Vol. 680, pp.12–23.

Song, X., Li, J., Tang, Y., Zhao, T., Chen, Y. and Guan, Z. (2021) 'JKT: a joint graph convolutional network based deep knowledge tracing', *Information Sciences*, Vol. 580, pp.510–523.

Takami, K., Flanagan, B., Dai, Y. and Ogata, H. (2024) 'Evaluating the effectiveness of Bayesian knowledge tracing model-based explainable recommender', *International Journal of Distance Education Technologies*, Vol. 22, No. 1, pp.1–23.

Wu, Z., Huang, L., Huang, Q., Huang, C. and Tang, Y. (2022) 'SGKT: session graph-based knowledge tracing for student performance prediction', *Expert Systems with Applications*, Vol. 206, pp.97–104.

Xia, Z., Dong, N., Wu, J. and Ma, C. (2023) 'Multivariate knowledge tracking based on graph neural network in ASSISTments', *IEEE Transactions on Learning Technologies*, Vol. 17, pp.32–43.

Xu, F., Chen, K., Zhong, M., Liu, L., Liu, H., Luo, X. and Zheng, L. (2024) 'DKVMN&MRI: a new deep knowledge tracing model based on DKVMN incorporating multi-relational information', *Plos ONE*, Vol. 19, No. 10, pp.22–35.

Ying, Z., Bourgeois, D., You, J., Zitnik, M. and Leskovec, J. (2019) 'GNNExplainer: generating explanations for graph neural networks', *Advances in Neural Information Processing Systems*, Vol. 32, pp.52–59.

Zhao, H. and Sun, Z. (2024) 'Traditional cultural network online education integrating deep learning and knowledge tracking', *Scalable Computing: Practice and Experience*, Vol. 25, No. 1, pp.341–354.

Zou, Y., Yan, X. and Li, W. (2020) 'Knowledge tracking model based on learning process', *Journal of Computer and Communications*, Vol. 8, No. 10, pp.7–17.