International Journal of Web-Based Learning and Teaching Technologies
Volume 19 • Issue 1

# DBGCN:
## A Knowledge Tracing Model Based on Dynamic Breadth Graph Convolutional Networks

Ping Hu, School of Information Engineering, Henan Institute of Science and Technology, China*
Zhaofeng Li, School of Information Engineering, Henan Institute of Science and Technology, China
Pei Zhang, School of Information Engineering, Henan Institute of Science and Technology, China
Jimei Gao, Faculty of Intelligent Engineering, Huanghe Jiaotong University, China
Liwei Zhang, School of Information Engineering, Henan Institute of Science and Technology, China

## ABSTRACT

Given the extensive use of online learning in educational settings, Knowledge Tracing (KT) is becoming increasingly essential. KT primarily aims to predict a student's future knowledge acquisition based on their past learning activities, thus enhancing the efficiency of student learning. However, the effective acquisition of dynamic and evolving student representations from their historical records presents a formidable challenge. This paper introduces a Knowledge Tracing methodology predicated on Dynamic Broadth Graph Convolutional Networks (DBGCN). DBGCN leverages the mechanisms of breadth graph convolutional networks to proficiently acquire representations of questions and knowledge points from dynamically constructed topological graphs. It employs student state information as an attention query vector to augment student representations, thereby partially mitigating the challenge of capturing the dynamic shifts in user states. The effectiveness of our proposed DBGCN method has been demonstrated through extensive experimentation.

## KEYWORDS

Dynamic Breadth, Graph Convolutional Networks, K-nearest Neighbor, Knowledge Tracing

## INTRODUCTION

Due to the rise of online education platforms, the trend towards intelligence in various online educational platforms, including massive open online courses (MOOCs), is becoming increasingly evident. Providing appropriate guidance based on learners' individual characteristics, such as strengths and weaknesses, can also help learners understand their learning progress. Knowledge tracking (Cui et al., 2022) aims to predict future knowledge acquisition of students based on their learning history, thereby enhancing learning efficiency. The research in this paper covers the issue of knowledge tracking in the field of education. Specifically, we focus on how to utilize students' historical learning data to predict their future learning needs and performance, and how to effectively track students' knowledge acquisition based on their learning behavior patterns and historical data

DOI: 10.4018/IJWLTT.342848
*Corresponding Author

to achieve personalized learning guidance. Knowledge tracking (KT) aims to accurately track how learners' understanding of concepts evolves over time, reflecting their past performance in exercises. This process forms the basis for subsequent tasks such as automated assessment of student abilities, rational planning of learning strategies, and accurate recommendation of exam resources.

Traditional KT methods, including Bayesian Knowledge Tracing (BKT) as proposed by Corbett and Anderson (2005), rely on a binary variable system to define each student's knowledge state. In this system, each variable indicates whether a student masters or does not master a specific knowledge point. Subsequently, it utilizes a Markov model derived from classical probability theory to gauge the student's level of knowledge mastery. Käser et al. (2017) propose a personalized BKT model approach that takes into account differences among students in two categories of model parameters. Nonetheless, the BKT model assumes that each question pertains to a single skill, and it treats different skills as independent entities. Consequently, these models are ill-suited for addressing problems that encompass multiple skills and are unable to capture the interconnections between distinct skill sets.

Over the past few years, motivated by the advancements in deep learning (Song et al., 2022), most recent research in knowledge tracing focuses on applying deep learning techniques. Piech et al. (2015) introduce DKT, which is an attempt to use Recurrent Neural Networks (RNN) —outlined by Sherstinsky (2020)— to model a student's practice history for predicting their performance. To track the complex nature of student learning, some studies have extended DKT by enhancing external memory structures, including through a Key-Value Memory Network (KVMN) (Miller et al., 2016). The latent variables used in this approach have stronger representation capabilities. Nonetheless, the static nature of their key-value matrices poses a challenge in the efficient monitoring of students' knowledge states.

Under the influence of the Transformer method by Cui et al. (2023), several research efforts have aimed to integrate graph attention mechanisms into KT. The fundamental concept revolves around acquiring the capacity to learn attention weights for questions within a student's learning history sequence. This addresses a constraint observed in the DKT model, which treats all questions with equal importance within a series of interactions. Ghosh et al. (2020) introduce a scaled dot-product attention networks in the KT model, learning student states from multiple subspaces (Ma et al., 2023). However, in knowledge tracing tasks, various relationship structures often exist, such as complex relationships between exercises and skills, as well as relationships between exercises. To capture these associations proficiently, the contemporary approach involves delving into graph network learning methodologies, such as Graph Neural Networks (GNNs) (Wan et al., 2023). Yang et al. (2021) proposes the Graph-based Interaction Knowledge Tracing (GIKT) method, which constructs a graph of question-skill relationships and uses GNNs to learn inter-graph relationships, allowing for more effective capture of relationships between sequences.

However, current research still needs to overcome the following two issues: (a) how to construct and capture relationships between features in a graph is crucial; and (b) how to effectively mine the hidden knowledge mastery in a student's historical interaction sequence with questions. Therefore, we propose a knowledge tracking model based on Dynamic Breadth Graph Convolutional Networks (DBGCN). It uses the K-nearest neighbor (KNN) method to construct a graph of question-skill relationships dynamically and employs a breadth search algorithm (BS) and graph convolutional networks (GCN) to capture relationships between questions and knowledge points from multiple modalities effectively. This reduces the impact of irrelevant questions when new questions interact with a student's historical question sequence. The experimental outcomes substantiate the efficacy of this methodology. To summarize, our principal contributions can be outlined as follows:

1. We introduce a knowledge tracking model rooted in the Dynamic Breadth Graph Convolutional Network to address the issue of capturing the user's evolving status efficiently.
2. We use the KNN method for constructing question-skill graphs and obtaining question and skill embeddings using dynamic breadth algorithms and graph convolutional networks.

3. This method reduces noise from irrelevant questions by allowing new questions to interact with a student's historical question sequence.

The organization of this paper is as follows. Related Works is an overview of pertinent literature. Next the paper outlines the findings and interpretation. It then presents the DBGCN model method, followed by experiments and evaluation, and then a discussion of the strengths and limitations of the DBGCN method. The final section concludes the entire work.

## RELATED WORKS

This section offers an overview of previous research within the domain of knowledge tracing, primarily focusing on two main categories of techniques: research based on Bayesian methods and research based on deep learning technologies.

### Bayesian-Based Knowledge Tracing Methods

Corbett and Anderson (2005) introduced the BKT method, which relies on a Hidden Markov Mode for time series data (Glennie et al., 2023). In the BKT model, a student's learning state is categorized as either "not mastered" or "mastered." However, these two learning states alone do not fully capture the dynamics of a student's learning process, as there exists an intermediate learning state between "not mastered" and "mastered." In this transitional learning state, a student may already have a grasp of a knowledge point. Recognizing the presence of this transitional learning state, the Three Learning States BKT (TLS-BKT) model introduced by Nakagawa et al. (2019) incorporates a three-way decision approach, dividing the learning process into three learning states. This model improves prediction accuracy relative to the BKT model and exhibits better robustness in statistical measures.

Qiu et al. (2011) introduced the KT-Forget model and the KT-Slip model, attempting to model the influence of time factors. This indicates that forgetting is the more likely cognitive explanation for the data. While the BKT method models each individual knowledge point separately, in reality, these knowledge points are not entirely independent; they are hierarchical and closely related. Agarwal et al. (2020) introduced the Multistate-Bayesian Knowledge Tracing model (MS-BKT) to address issues in classical knowledge tracing models, such as constant learning rates and only two knowledge states. By extending the knowledge states from "not learned" and "learned" to 21 different states, the addition of multiple states allows for a more precise assessment of student learning states, enhancing model performance.

Many researchers have proposed extended models based on Bayesian knowledge tracing. While these improved models have achieved some success, the fundamental assumptions of the BKT model have inherent limitations. The one-to-one correspondence between variables and knowledge components (KC) cannot be established, and the model itself loses crucial information during data processing. Consequently, it cannot accurately simulate a student's knowledge state. As a result, the BKT model faces challenges in widespread adoption in practical teaching scenarios.

### Deep Learning-Based Knowledge Tracing Models

While Bayesian-based models excel in modeling student knowledge states, the learning and cognitive processes of students are influenced by numerous micro factors that are challenging to capture using these models. Although it demonstrates strong performance, the direct inference of a student's knowledge mastery from the model's hidden states is a challenging task. This limitation makes it difficult for the model to depict a student's genuine knowledge state accurately. To address this, certain research efforts have extended Dynamic Keyhole Tracing (DKT). Xiong et al. (2016) introduced Extended-Deep Knowledge Tracing, which enhances DKT by incorporating supplementary student features. Yeung and Yeung (2018) proposed KT+ to address issues such as the problem of reconstructing input observations in DKT and the problem of waveform transitions in predictions. This approach is applicable for intelligent course design, enabling direct interpretation and discovery of the structure of student tasks. Based on the DKT model, which uses RNN to capture long-distance dependencies, Sun et al. (2019) show that DKVMN has the capability to utilize the interconnections among knowledge points to directly predict a student's mastery level for each individual knowledge point. This significantly enhances the accuracy of predictions. Shuang et al. (2020) designed hierarchical convolutional layers to extract personalized learning rates based on students' continuous learning interactions. The ASSIST2009 dataset is used in this study. However, in previous methods, models mostly simplify exercise records into knowledge sequences, thus transforming the task into a time-related prediction problem. From a data structure perspective, there may be rich dependencies between knowledge points.

Considering the effectiveness of Graph Neural Network methods (Zeng et al., 2022), some research attempts to incorporate them into knowledge tracing (Jiang et al., 2022). The innovation lies in extracting several factors effective in modeling exercise difficulty and simulating students' learning processes more realistically. However, constructing the graph structure is a challenge since the graph structure itself, including relevant concepts and relationships, lacks explicit values. Liu et al. (2023) introduced a novel SGD proximal algorithm for GCNs with an inexact operator, effectively quantifying the trade-off between smoothness and sparsity in GCN by analyzing the stability of the SGD proximal algorithm. Li et al. (2023) utilized two separate encoders (online encoder and target encoder) to encode two augmented versions of the user-item bipartite graph. To facilitate interaction between these two distinct graph encoders, they introduced local and global regularization, constructing positive and negative sample pairs. By replacing the filters in the convolutional layer with random weights and simultaneously adjusting the learning objective to use regularized least squares loss, Huang et al. (2022) investigated the theoretical and empirical research on GCN with random weights. The effectiveness and efficiency of the GCN-RW model in semi-supervised node classification tasks were verified. Fang et al. (2023) introduced the Graph-based Knowledge Tracing (GKT) model, which offers three approaches based on the adjacency matrix, multi-head attention mechanism, and variational autoencoder. Moreover, using only KC as input overlooks issues related to multiple KCs and the interactions between exercises. Kukkar et al. (2023) introduced PEBG, which uses embedded vector values for training to enhance the accuracy of KT. AKT is used entirely on attention networks, utilizing a novel monotonic attention mechanism (Ghosh et al., 2020). However, it also has limitations as the attention layer of the model is too shallow, making it challenging to capture the complex relationships that exist between different exercises and cognitive states. Nakagawa et al. (2019) implemented two methods proposed by Kudryashov to extract optical soliton solutions for the concatenation model. This concatenation model was a combination of the nonlinear Schrödinger equation, the Lakshmanan-Porsezian-Daniel model, and the Sasa-Satsuma equation. A full spectrum of soliton solutions emerged, along with comprehensive presentation of parameter constraints. KSGAN employed GAT to capture high-order relationships from the problem and knowledge skill graph (Mao et al., 2022). Additionally, it designed an integration function to learn problem representations and optimize the loss function. However, the structure of the problem-knowledge skill graph remains static.

Song et al. (2022) proposed the Bi-CLKT model, which designs a graph of relationships between exercises and utilizes a joint contrastive loss. The model employs RNN and an enhanced neural network as a prediction layer to obtain improved representations of exercises and concepts. However, it randomly selects edges or nodes for deletion to create new sets of nodes and edges. Nevertheless, this approach may inadvertently lead to the deletion of nodes strongly associated with the source node. Zeng et al. (2022) introduced the Difficulty-aware Convolutional Knowledge Tracing (DACK) model, which adeptly extracts multiple factors for effectively modeling exercise complexity and finely simulating shifts in student knowledge states and aptitude. It effectively emulates students' learning processes with greater realism and demonstrates strong performance when applied to real-world datasets. Jiang et al. (2022) proposed the Multichannel Attention Networks-based KT model (MAKT), introducing constraint factors to extract the relationship matrix between concepts and exercises and developing a KT model based on multi-channel attention networks. Considering the dependencies between exercises, Yang et al. (2021) introduced GIKT, which utilizes a GCN network to aggregate exercise-KC embedded features learned from high-order relationships. It directly combines exercise embeddings with the corresponding answer embeddings as inputs to the KT model. However, it employs a static graph structure.

It can be observed that current graph-based research predominantly utilizes static graph structures, making it challenging to adjust the graph structure flexibly based on real-world scenarios. Therefore, this paper proposes the DBGCN model, employing the KNN method to construct a dynamic problem-knowledge point topology graph. By connecting each node with its K-nearest neighbors, the model achieves flexible adjustments to the graph structure. By combining a breadth-first search algorithm to aggregate closely connected neighboring nodes, introducing an attention network to focus selectively on crucial information, and ultimately using the results from the GRU network as embedding vectors for the attention network, the model emphasizes consideration of temporal information.

By synthesizing these methods, DBGCN successfully learns node information at different levels and time sequences in knowledge tracking tasks, significantly improving prediction performance.

## Findings and Interpretation

In this section, we introduce the process of constructing the knowledge skill graph. Assume that the set of problems is represented as $Q = \{q_1, q_2, ..., q_m\}$ and the set of knowledge points is represented as $S = \{s_1, s_2, ..., s_n\}$. The vector representations for problems and knowledge points are denoted as $q_i$ and $s_j$, respectively.

Firstly, cosine similarity is employed to calculate the similarity between problems and knowledge points.

$$c_{ij} = cosin(q_i, s_j) \tag{1}$$

For each question $q_i$, calculate its similarity with all relevant knowledge points, resulting in a list of similarities. Select the top K knowledge points with the highest similarities, i.e., choose the top K neighboring nodes:

$$KNN(q_i) = topk(\{c_{i1}, c_{i2}, ..., c_{in}\}, K) \tag{2}$$

For each question $q_i$, establish edges between the question and the selected top *K* neighboring nodes $KNN(q_i)$, forming a problem-knowledge point topology graph $\mathcal{G}$. This can be represented as:

$$Edge\left(q_i, s_j\right) = \begin{cases} 1, & if \;\; s_j \in KNN\left(q_i\right) \\ 0, & otherwise \end{cases} \tag{3}$$

For each pair of question $q_i$ and neighboring node $s_j$, the edge weight will be represented by the similarity value. This kind of graph structure is more flexible, allowing for the inclusion of more information in the graph.

**Table 1. Dataset**

| | ASSIST2009 | ASSIST2012 | Algebra2006 |
|---|---|---|---|
| #Students | 3,241 | 29,018 | 1,130 |
| #Questions | 17,709 | 53,086 | 129,263 |
| #Concepts | 124 | 265 | 550 |
| #Records | 278,868 | 2,711,602 | 1817,393 |

## MATERIALS AND METHODS

### Materials

We conduct experiments in the knowledge tracking domain using three extensive publicly accessible datasets: ASSIST2009, ASSIST2012, and Bridge Algebra 2006. The data set refers to the paper x settings. Comprehensive dataset statistics can be found in Table 1.

The ASSIST2009 dataset is based on the Item Response Theory (IRT) model and consists of binary responses from students on each item, typically indicating correct or incorrect. It includes data such as student ID, item ID, student response, and item parameters. It is widely used in research on knowledge tracing and learning assessment.

Similar to ASSIST2009, the ASSIST2012 dataset is also based on the IRT model and records binary responses from students on each item. It contains information such as student ID, item ID, student response, and item parameters. This dataset is valuable for assessing students' knowledge levels and skills mastery, as well as researching the effectiveness of teaching strategies.

The Algebra2006 dataset focuses on algebra learning and records sequences of student operations in solving algebraic equations. Each student's response is represented as a series of steps, including student ID, step ID, student operation, and step parameters. This dataset is used for analyzing student problem-solving strategies, identifying learning difficulties, and proposing personalized teaching methods.

### Methods

#### *Graph Convolutional Network*

A GCN is a type of neural network structure based on graphs, designed specifically for handling graph-structured data. It is a deep learning model aimed at learning representations of nodes in a graph. GCN propagates information along the edges of the graph, aggregating information from the neighbors of nodes, in order to update the representations of nodes. This manner of information aggregation and updating allows GCN to learn relationships between nodes and generate meaningful representations, enabling nodes to be used for various tasks such as node classification, link prediction, etc.

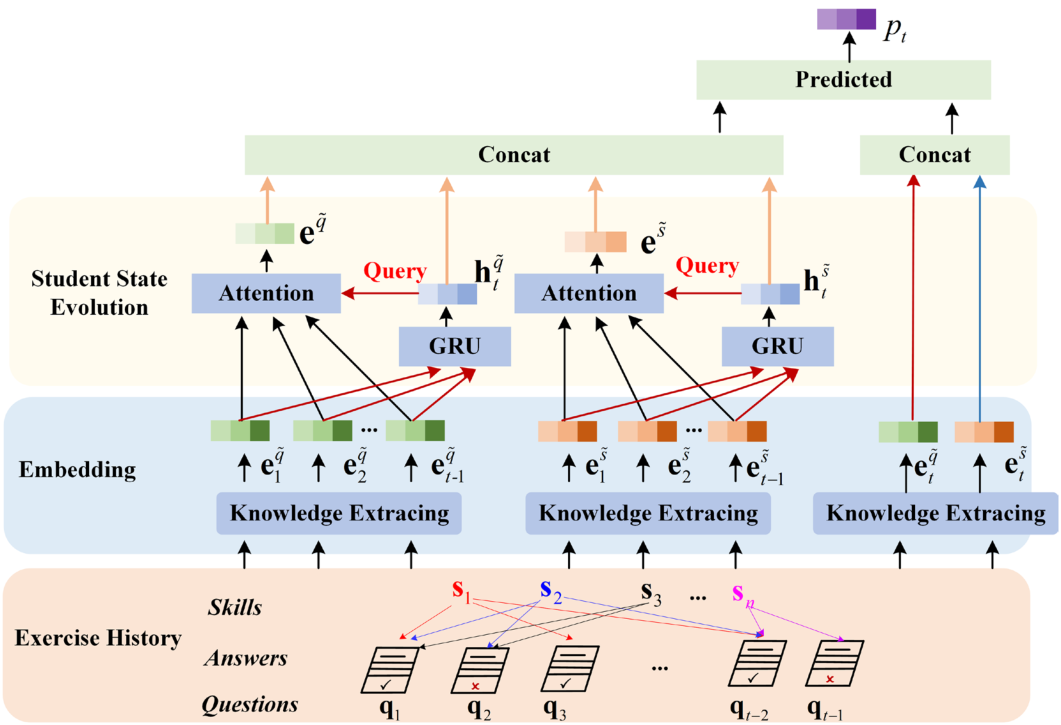
#### *Attention Network*

An Attention Network is a type of neural network structure designed to enhance the model's focus on input data. The model assigns different attention weights to different parts of the input data, enabling it to handle input data more flexibly and adapt better to various tasks and scenarios.

## PROPOSED METHODOLOGY

The overall architecture of this framework, as shown in Figure 1, is primarily divided into four key modules, including the Exercise History Layer, Embedding Layer, Student State Evolution Layer, and Prediction Layer. Using the K-nearest neighbor (KNN) method dynamically constructs graph

**Figure 1. Overall Framework of DBGCN**




topology, emphasizing relationships between knowledge points. This enhances the ability to track student knowledge levels. At the same time, the breadth-first graph convolutional network is utilized to aggregate neighboring nodes in the problem and knowledge point graphs, better adapting to individual learning characteristics. By utilizing the student state information obtained from GRU as the query vector for the attention network, we successfully learn the student state information, improving the accuracy of tracking student learning states.

## Embedding

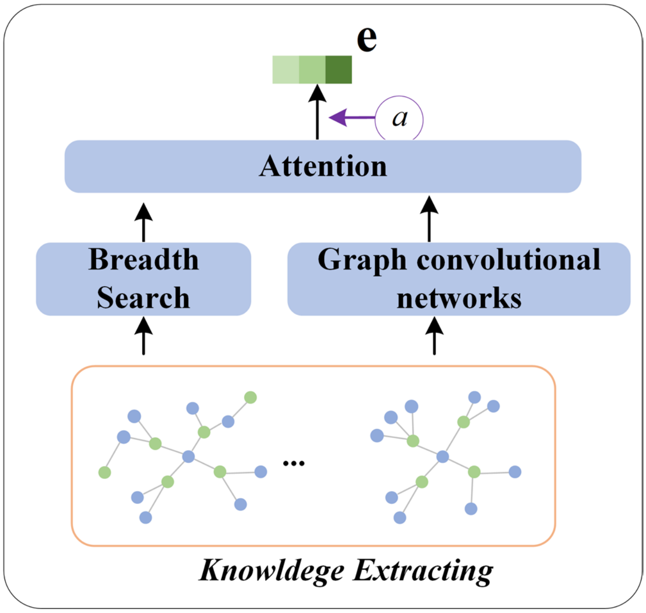
The DBGCN method utilizes the Embedding method to obtain embedding vectors for questions, knowledge points, and answers (Fang et al., 2023). The resulting vectors are denoted as $E_s \in \mathbb{R}^{|S| \times d}$ and $E_a \in \mathbb{R}^{2 \times d}$, with $d$ representing the embedding dimension.

## Knowledge Extracting

Given the sparsity of question data, especially in cases with very limited training samples, effectively representing information-rich questions poses a significant challenge. By establishing a question-knowledge point relationship graph $\mathcal{G}$, it is possible to mitigate the problem of data sparsity and obtain a more effective representation of questions $E_q$.

The first step involves establishing a question-knowledge point graph structure in the student's historical interaction exercise sequence. Traditional methods typically treat the constructed graph structure as static. However, in DBGCN, we employ the K-Nearest Neighbors (KNN) method, continuously recalculating the nearest neighbors of each node in the feature space at each layer, achieving dynamic construction of the graph structure $\mathcal{G}$.

**Figure 2. Knowledge Extraction Layer Framework**




Next, the DBGCN method utilizes knowledge embedding techniques to obtain representations of questions and knowledge points. As illustrated in Figure 2, we use breadth-first search algorithms and graph convolution networks to capture deep relationships between questions and knowledge points or knowledge points and questions. Finally, we apply the Concatenation operation to obtain more effective representations of questions or knowledge points (Kukkar et al., 2023).

Graph convolution networks stack multiple graph convolution layers to acquire high-order neighbor information; each layer updates nodes based on themselves and their adjacent nodes. Suppose the representation of the $i$-th node in the graph is denoted as $Z_i^{gcn}$ (which can represent the embedding vector of a knowledge point or a question), the embedding vector of that node is obtained by weighted averaging of the information from its neighbors and itself. The GCN formula is as follows:

$$\hat{A} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} X \tag{4}$$

$$Z_i = f(X, A) = \mathrm{softmax}\left(\hat{A}\,\mathrm{ReLU}\left(\hat{A} X W^{(0)}\right) W^{(l)}\right) \tag{5}$$

where $\tilde{A} = A + I_N$ and $I_N$ are N-dimensional identity matrices and $A$ is the adjacency matrix.

An effective neighbor information aggregation mapping function plays a crucial role in identifying relevant neighborhood information for a given node in a graph while filtering out irrelevant information. Selecting the most relevant neighboring nodes with respect to the current node is a critical problem. To effectively alleviate this issue, we propose a breadth-first search algorithm that filters out neighbors more closely connected to the current source node. Furthermore, it explores the neighbors of these selected nodes, thus achieving effective aggregation of neighborhood node information. First, we calculate the weights of the first-order neighbor nodes, which are used in this process:

$$\alpha(x,y) = softmax(\nu^T \tanh(W_s^T x + W_d^T y + b)) \tag{6}$$

where $W_s$ is the weight of the source node and $W_d$ is the weight of neighboring nodes.

Then, the neighboring nodes are subjected to weighted processing while incorporating the node's own information:

$$x_i = \tanh\left(W^T \sum_{j \in \mathcal{N}_{(i)\cup\{i\}}} \alpha\left(x_i, x_j\right) \cdot x_i\right) \tag{7}$$

where $x_i$ represents the *i*-th node, and $\mathcal{N}_i$ represents the set of neighboring nodes for the *i*-th node. To facilitate this, we will denote $x_i$ as $Z_i^{bs}$.

To capture richer relationships between nodes, the node embedding vectors obtained from both methods are fused by concatenating them together to obtain the final representation of the *i*-th node as follows:

$$e_i = Relu(W_1([Z_i^{gcn}, Z_i^{bs}]) + b_1) \tag{8}$$

Subsequently, the question and skill embeddings are integrated to evaluate the student's proficiency in the current question or knowledge points. We use $e_i^{\tilde{q}}$ and $e_i^{\tilde{s}}$ to represent the embedding vectors of questions and knowledge points at any given time *t* in the student's interaction history, where $x$ can be either $e_i^{\tilde{q}}$ or $e_i^{\tilde{s}}$:

$$e = \sigma(W([x,a]) + b) \tag{9}$$

## Student State Evolution

In this layer, the objective is to learn accurate representations of student states from dynamically changing student data. Information about a student's state is carried within their sequence of interactions with questions. Each time a student completes a question, their individual knowledge state undergoes slight changes. To address the challenge of modeling these latent knowledge states, Gated Recurrent Units (GRUs) are introduced to capture dependencies between actions. The input to the GRU is the student's question interaction data ordered over time, and its hidden state is used to represent the student's knowledge state, reflecting the student's level of knowledge mastery.

First, obtain the student's state, denoted as $h_t^{\tilde{q}}$, from the student's history. Then, use this state as the query vector for attention. This approach has the advantage of increasing the weights associated with exercise or knowledge points related to the student's state. Finally, through the attention network aggregation, more comprehensive representations of the student's historical questions and knowledge points, denoted as $e_a^{\tilde{q}}$ and $e_a^{\tilde{s}}$, are obtained:

$$e_a^{\tilde{q}} = Attention(GRU(e_i^{\tilde{q}}, h_{i-1}^{\tilde{q}}), e_i^{\tilde{q}}, e_j^{\tilde{q}}) \tag{10}$$

$$e_a^{\tilde{s}} = Attention(GRU(e_i^{\tilde{s}}, h_{i-1}^{\tilde{s}}), e_i^{\tilde{s}}, e_j^{\tilde{s}}) \tag{11}$$

Next, concatenate the questions $e_a^{\tilde{q}}$ and knowledge points $e_a^{\tilde{s}}$ that the student has historically learned, along with the student's mastery states $h_{i-1}^{\tilde{q}}$ and $h_{i-1}^{\tilde{a}}$, to get the final representation of the student's state denoted as $u$:

$$u = [e_a^{\tilde{q}}, e_a^{\tilde{s}}, h_{i-1}^{\tilde{q}}, h_{i-1}^{\tilde{s}}] \tag{12}$$

Concatenate the new question and knowledge points at time *t* to obtain $v$, and then feed it along with the student representation $u$ into the prediction module:

$$v = [e_t^{\tilde{q}}, e_t^{\tilde{s}}] \tag{13}$$

### Click to Predict

Compute the inner product between the student representation and the new question representation in order to derive the prediction probability $p_t$:

$$p_t = sigmoid(u, v) \tag{14}$$

### Optimization

We utilize the gradient descent method to update the model parameters iteratively. This is accomplished by minimizing the cross-entropy loss function, which measures the disparity between the predicted probability of a correct answer and the actual student response labels:

$$\mathcal{L} = -\sum_t (a_t \log p_t + (1 - a_t) \log(1 - p_t)) \tag{15}$$

### Flowchart

Figure 3 shows a flowchart of the DBGCN method.

### Experimental Results

In this research, we carried out a lot of experiments with the objective of evaluating the effectiveness of our model. We conducted comparisons between our model and other baseline approaches using three publicly available datasets to gauge predictive performance and address the following three inquiries:

1. How does the performance of the DBGCN method stack up against the current state-of-the-art methods?
2. What impact do key model designs in DBGCN have on the experimental results?
3. How does the influence of hyperparameter settings affect the experimental results?

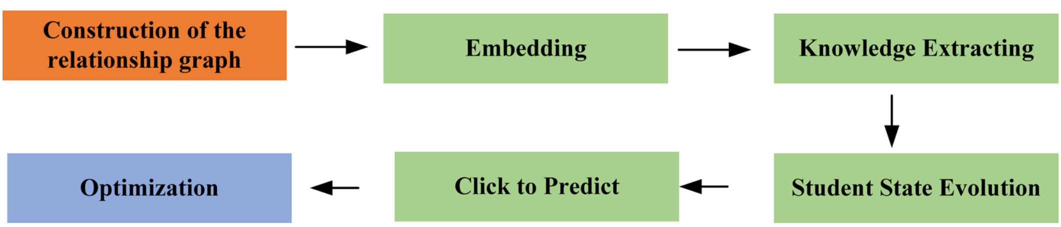
### Evaluation

We utilize the Area Under the Receiver Operating Characteristic curve (AUC) as an evaluation metric, with values typically ranging between 0.5 and 1.0. A higher AUC value indicates better modeling performance (Nguyen et al., 2023).

**Algorithm 1. DBGCN**

```
1:   Initialize:
     Weights; bias parameters;
     n:epoch; m: length of the student sequence
     Begin:
2:   For epoch in n do
3:     For j in m do
4:       Construct a graph of problem and skills 𝒢
5:       Get embedding vectors e_i^q̃ and e_i^s̃
6:     End for
7:     Get the student state h_t^q̃
8:     Get the final student state u
9:   Compute the probability p_t
10:    Update parameters
11:  End For
```

**Figure 3. The Flowchart of DBGCN**




## Baselines

Currently, advanced baseline models are as follows. First, GKT (Nakagawa et al., 2019). The first application of graph neural networks in the field of knowledge tracking involves constructing a graph of KC relationships. It reframes the knowledge tracking task as a time-series node-level classification problem in GNN, enabling improved prediction accuracy without relying on additional information. However, the GKT model only utilizes KC as input, overlooking issues such as the influence of multiple KCs and interactions between exercises.

Next is AKT (Ghosh et al., 2020). It entirely relies on attention networks, employing a novel monotonic attention mechanism. It combines the history of answer records and the future response to assessment questions with an interpretable model. Attention weights are calculated using exponential decay and context-aware relative distance measures. Although utilizing attention networks has certain advantages, it also has limitations. The model's attention layer is too shallow, making it challenging to capture the complex relationships present in different exercises and cognitive states.

Another model is GIKT(Yang et al., 2021). Inspired by the powerful capabilities of GNN in aggregating neighbor information to extract graph representations, we utilize GCN networks to aggregate exercise-KC embedded features learned from high-order relationships. We directly use exercise embeddings along with corresponding answer embeddings as inputs to the KT model. Additionally, we design a reformulation module and an interaction module that can consistently better simulate students' mastery of new exercises and their related KCs.

One further model is DACK (Zeng et al., 2022). In previous works, the majority focused on modeling features related to questions or knowledge points. However, there are many other features such as student pass rates, attempt counts, and time spent that can enrich the model's learnable features by modeling the difficulty of exercises. Additionally, dividing the student's historical interaction sequence into two matrices allows for the extraction of micro-changes in the student's knowledge state and cognitive ability. Finally, the introduction of the Rasch psychometric measurement model enhances predictive performance.

The final model is MAKT (Jiang et al., 2022). To overcome the limitation of prior works that only utilized concepts or exercises as inputs for training, we introduce constraint factors to extract the relationship matrix between concepts and exercises. We propose extracting the relationship matrix by constraining co-occurrence relationships through a time window.

## Comparative Analysis

All methods are implemented using the TensorFlow framework in the present study. The embedding dimension is 100. The maximum aggregation layer is set to "$l = 2$." Dropout is set to 0.8 (Giesemann et al., 2023). The optimized algorithm is Adam (Reyad et al., 2023), the learning rate is 0.001, and the mini-batch size is configured as 32. We obtained the values for other parameters through multiple experiments, such as the value of K for KNN. The training dataset constitutes 80% of the data, while the testing dataset accounts for the remaining 20%.

## The Comparative Results With Baseline Models (RQ1)

According to Table 2, the DBGCN model demonstrates high performance on most indicators across the three datasets. The GKT model, despite utilizing graph neural networks, exhibits a relatively simple graph structure by solely using KC to construct the graph. Consequently, it struggles to learn rich information from the graph, leading to subpar model performance. AKT, employing attention networks to effectively capture relationships between historical answer records, faces limitations due to a shallow attention layer, making it challenging to grasp complex relationships among different exercises and cognitive states. GIKT leverages GCN to learn information from the question and knowledge point graph structures, resulting in improved performance compared to the GKT model. However, its graph structure of question-knowledge point pairs is static. MAKT uses constraint factors to extract the relationship matrix between concepts and exercises, effectively learning the

**Table 2. The AUC Results Over Three Datasets**

| Method | ASSIST2009 | ASSIST2012 | Algebra2006 |
|---|---|---|---|
| GKT | 0.743 | 0.745 | 0.748 |
| AKT | 0.783 | 0.772 | 0.737 |
| GIKT | 0.779 | 0.771 | 0.778 |
| MAKT | 0.782 | 0.774 | 0.781 |
| DACK | 0.785 | 0.773 | 0.783 |
| **DBGCN** | **0.792** | **0.775** | **0.788** |

relationships and enhancing the model's capability to extract information representations through graph generation learning, graph contrastive learning, and matrix factorization. DACK considers additional features, such as question difficulty, to enrich the learnable features. Additionally, it divides the student's historical interaction sequence into multiple matrices and introduces a psychological model to enhance the model's predictive ability effectively.

In contrast, DBGCN outperforms other baseline models. The reason is its dynamic construction of the problem-knowledge point topology graph using the KNN method, allowing the model to flexibly adjust the graph structure based on real-world scenarios. Other models, in comparison, may adopt static or fixed graph structures. By employing GCN networks and breadth-first search algorithms to filter and aggregate node neighborhood features, and finally combining attention networks and GRU networks, the model effectively captures changes in a student's knowledge over different time steps. This combination likely enhances the modeling capability for temporal information. Through the comprehensive use of graph convolutional networks, breadth-first search algorithms, attention networks, and GRU networks, DBGCN can efficiently learn node information at different levels and time steps, thereby enhancing predictive performance in knowledge tracking tasks. Therefore, the DBGCN model exhibits superior predictive performance.

## Ablation Studies RQ2

### *Effect of KNN Layer*

To explore whether the use of KNN in constructing the graph structure is beneficial for model learning, we conduct the following experiment. In the experiment, DBGCN-NO-KNN represents the model without the use of the KNN method. As shown in Figure 4, DBGCN exhibits better performance. This is because in the initial stages, there is a lack of topological information between questions and knowledge points, making it challenging to represent their relationships effectively. Associations between nodes have not been established. Therefore, the use of the KNN method to create topological relationships between nodes enhances the node representation capability. Furthermore, DBGCN can dynamically construct the graph structure since it can aggregate the top $K$ nodes most relevant to the source node by setting different $K$ values as needed.

### *Impact of the Breadth Graph Convolutional Network Structure*

To explore whether the proposed breadth graph convolution network module contributes to the model's predictions, we conducted the following experiment. In the experiment, "DBGCN-No-Bread" represents the model without the breadth-first search algorithm, while "DBGCN-No-GCN"

**Figure 4. Results with and without KNN Method**

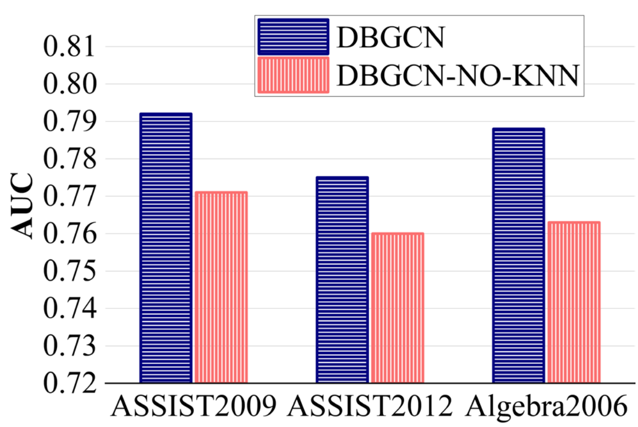
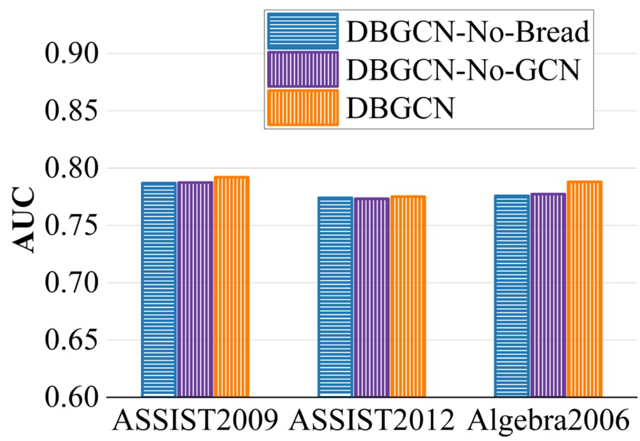


**Figure 5. The Performance of the Breadth Graph Convolutional Network**



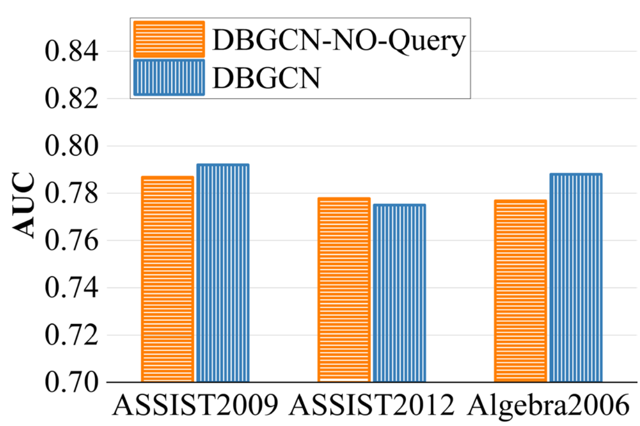

represents the model without the graph convolution network algorithm, and we also included our proposed DBGCN method. As seen in Figure 5, DBGCN outperforms the others. For graph-based algorithms, their essence lies in learning a mapping function that effectively aggregates neighborhood information to filter out irrelevant data.

The breadth-first search algorithm helps to select the closest neighboring nodes to the current node, while the graph convolution network (GCN) aggregates neighborhood nodes from a global perspective. When these two methods are combined, they can aggregate neighborhood information from both local and global perspectives, effectively improving prediction accuracy.

### *Impact of the Presence of Query Vectors*

To explore whether using user state features obtained from GRU as query vectors in the attention network has an impact on the experiments, we conducted the following experiment, where "DBGCN-NO-Query" represents not using query vectors. As shown in Figure 6, using DBGCN (with query vectors) yields better results. This is because the user state information obtained from GRU reflects the student's recent mastery of knowledge and skills. Using this state information as query vectors in the attention network increases the weight associated with questions or knowledge points that are

**Figure 6. The Performance of the Presence or Absence of Query Vectors**



relevant to what the student currently knows, thereby more effectively capturing the student's current learning state.

## Hyperparameter Analysis (R3)

### *Impact of Different K*

To explore the impact of the value of *K* in the KNN method, we tested different numbers of nearest neighbors, including [10, 20, 30, 40]. As shown in Figure 7, it can be observed that performance decreases when *K* is larger. This is because a larger *K* value can lead to the aggregation of some noisy nodes, affecting the model's performance. Therefore, in this experiment, we chose a *K* value of 20 as it demonstrated a good balance in terms of performance.

### *Effect of the Number of Graph Convolution Network Layers*

To explore the impact of the number of layers (l) in the graph convolution network, we conducted the following experiment, trying different values of *l*, with *l*= [1, 2, 3, 4]. As shown in Figure 8, the model

**Figure 7. The Impact of Different K Values on the Experiments**

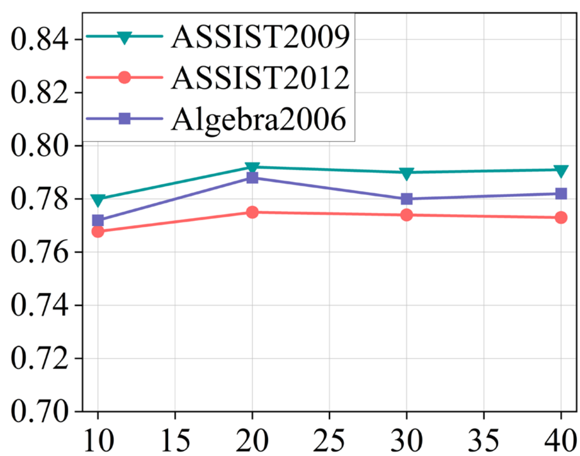
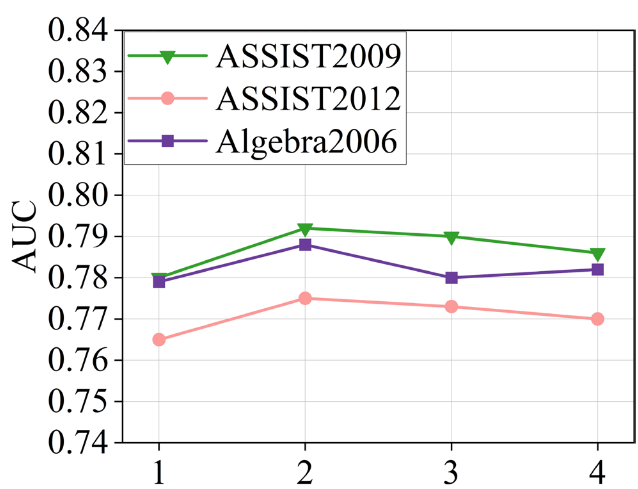



**Figure 8. The Results of Different Graph Convolution Network Layer Numbers**



performs best when $l$ is 2. A single layer of GCN can handle information from first-order neighboring nodes in the graph, and multiple layers of GCN are typically needed to aggregate information from $l$th-order neighboring nodes. However, too many layers can result in an excessive number of aggregations of information from each vertex to its surrounding neighboring nodes. This situation can lead to all vertices becoming increasingly similar, eventually converging to similar values, making it difficult to distinguish the individual features of each vertex. Therefore, in this experiment, choosing a layer value of $l$=2 appears to strike a good balance between performance and model complexity.

### *Impact of Embedding Dimensions on the Experiments*

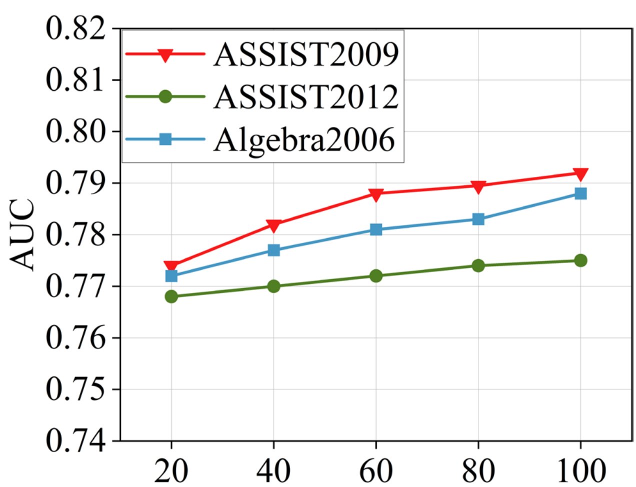
To explore the impact of embedding dimensions on the experiments, we conducted the following experiment, setting the dimensions of question and knowledge embeddings, $d$, to [20, 40, 60, 80, 100], to study the effect of changing the embedding dimension on model performance. As shown in Figure 9, as the embedding dimension increases, the predictive performance of DBGCN gradually improves. However, once the embedding dimension reaches a certain value, the change in AUC becomes small. In this experiment, we chose to set the embedding dimension $d$ to 100.

## Discussion

### *Advantages*

We found that the DBGCN serves as an effective method for knowledge tracking, demonstrating extensive potential applications in the learning process. By dynamically constructing graph topology using the KNN method, we highlight the connections between knowledge points, leading to a notable enhancement in tracking students' knowledge levels. Moreover, utilizing breadth-first graph convolutional networks to aggregate neighboring nodes in both problem and knowledge point graphs enables us to better accommodate individual learning characteristics. Crucially, by integrating student state information acquired from GRU as the query vector for the attention network, we effectively capture student state details and achieve significant improvement in accurately monitoring student learning states. These findings suggest that the DBGCN method holds significant potential in the field of knowledge tracking, providing valuable guidance for future research and practice.

**Figure 9. The Results of Embedding Dimensions**



### *Limitations*

Building the topological graph structure for questions and knowledge points effectively enhances the capability to obtain question and student representations. However, there are many other features to consider, such as text features, especially with the emergence of modern large-scale natural language processing models. This approach becomes more feasible, as well as image features and so on. How to supplement these additional features and employ more effective methods to process them will be a focus of our future work.

## CONCLUSION

In this study, we introduced Dynamic Broad Graph Convolutional Network (DBGCN) as an effective method for knowledge tracking and discussed its application in the learning process. DBGCN utilizes the KNN method to construct the graph topology dynamically, emphasizing the relationships between knowledge points. This thereby enhances the granularity of tracking student knowledge levels. Additionally, we employed a breadth-first graph convolutional network to aggregate neighboring nodes in the problem and knowledge point graphs, better adapting to individual learning characteristics. By leveraging the student state information obtained from GRU as the query vector for the attention network, we successfully learned student state information, thereby improving the accuracy of predicting student learning states.

Despite achieving certain accomplishments, we recognize some limitations in the current method, such as the inadequate utilization of other key features like text and image features, as well as the issue of high model algorithm complexity. At the same time, there is also the problem of insufficient application scenarios. We plan to consider more application scenarios in the future to improve the applicability of the model. Future research should prioritize addressing these issues, such as expanding datasets to validate the model's universality, in order to further advance and practically apply knowledge tracking in the field.

## CONFLICTS OF INTEREST

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

## FUNDING STATEMENT

This research was supported by the Henan Province Science and Technology Research Project [grant number 222102320111, 222102320218].

## PROCESS DATES

Received: November 3, 2023, Revision: March 22, 2024, Accepted: March 11, 2024

## CORRESPONDING AUTHOR

Correspondence should be addressed to Ping Hu (hp2014qw@163.com)

## REFERENCES

Agarwal, D., Baker, R. S., & Muraleedharan, A. (2020). *Dynamic knowledge tracing through data driven recency weights* [Paper presentation]. At the International Conference on Educational Data Mining, Online.

Corbett, A. T., & Anderson, J. R. (2005). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, *4*(4), 253–278. doi:10.1007/BF01099821

Cui, C., Zong, J., Ma, Y., Wang, X., Guo, L., Chen, M., & Yin, Y. (2022). Tri-branch convolutional neural networks for top-k focused academic performance prediction. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(1), 1–12. PMID:35588409

Cui, J., Chen, Z., Zhou, A., Wang, J., & Zhang, W. (2023). Fine-grained interaction modeling with multi-relational transformer for knowledge tracing. *ACM Transactions on Information Systems*, *41*(4), 1–26. doi:10.1145/3580595

Fang, H., Wang, Y., Tian, Z., & Ye, Y. (2023). Learning knowledge graph embedding with a dual-attention embedding network. *Expert Systems with Applications*, *212*, 118806. doi:10.1016/j.eswa.2022.118806

Ghosh, A., Heffernan, N., & Lan, A. S. (2020). Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* Association for Computing Machinery. doi:10.1145/3394486.3403282

Giesemann, J., Delgadillo, J., Schwartz, B., Bennemann, B., & Lutz, W. (2023). Predicting dropout from psychological treatment using different machine learning algorithms, resampling methods, and sample sizes. *Psychotherapy Research*, *33*(6), 683–695. doi:10.1080/10503307.2022.2161432 PMID:36669124

Glennie, R., Adam, T., Leos-Barajas, V., Michelot, T., Photopoulou, T., & McClintock, B. T. (2023). Hidden Markov models: Pitfalls and opportunities in ecology. *Methods in Ecology and Evolution*, *14*(1), 43–56. doi:10.1111/2041-210X.13801

Huang, C., Li, M., Cao, F., Fujita, H., Li, Z., & Wu, X. (2022). Are graph convolutional networks with random weights feasible? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(3), 2751–2768. doi:10.1109/TPAMI.2022.3183143 PMID:35704541

Jiang, X., Ouyang, Y., Liu, Z., Rong, W., & Xiong, Z. (2022). MAKT: Multichannel attention networks based knowledge tracing with representation learning. In *Proceedings of 2022 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE).* IEEE. doi:10.1109/TALE54877.2022.00055

Käser, T., Klingler, S., Schwing, A. G., & Gross, M. (2017). Dynamic Bayesian Networks for student modeling. *IEEE Transactions on Learning Technologies*, *10*(4), 450–462. doi:10.1109/TLT.2017.2689017

Kukkar, A., Kumar, S., Malik, S., Biswas, A., Yıldırım, Y., Moshokoa, S. P., Khan, S., & Alghamdi, A. A. (2023). Optical solitons for the concatenation model with Kurdryashov's approaches. *Ukrainian Journal of Physical Optics*, *24*(2), 155–160. doi:10.3116/16091833/24/2/155/2023

Li, M., Zhang, L., Cui, L., Bai, L., Li, Z., & Wu, X. (2023). BLoG: Bootstrapped graph representation learning with local and global regularization for recommendation. *Pattern Recognition*, *144*, 109874. doi:10.1016/j.patcog.2023.109874

Liu, S., Wei, L., Lv, S., & Li, M. (2023, August). Stability and generalization of $\ell$pregularized stochastic learning for GCN. *In Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence,* 5685-5693.

Ma, T., Wang, W., & Chen, Y. (2023). Attention is all you need: An interpretable transformer-based asset allocation approach. *International Review of Financial Analysis*, *90*, 102876. doi:10.1016/j.irfa.2023.102876

Miller, A., Fisch, A., Dodge, J., Karimi, A. H., Bordes, A., & Weston, J. (2016). Key-value memory networks for directly reading documents. In *Proceedings of 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. doi:10.18653/v1/D16-1147

Nakagawa, H., Iwasawa, Y., & Matsuo, Y. (2019). Graph-based knowledge tracing: Modeling student proficiency using graph neural network. In *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence.* IEE. doi:10.1145/3350546.3352513

Nguyen, N. D., Tan, W., Du, L., Buntine, W., Beare, R., & Chen, C. (2023). AUC maximization for low-resource named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence.* AAAI Press. doi:10.1609/aaai.v37i11.26571

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. (2015). Deep knowledge tracing. In *Proceedings of Annual Conference on Neural Information Processing Systems*. MIT Press.

Qiu, Y., Qi, Y., Lu, H., Pardos, Z. A., & Heffernan, N. T. (2011). *Does Time matter? Modeling the effect of time with Bayesian Knowledge Tracing* [Paper presentation]. At the International Conference on Educational Data Mining, Eindhoven, The Netherlands.

Reyad, M., Sarhan, A. M., & Arafa, M. (2023). A modified Adam algorithm for deep neural network optimization. *Neural Computing & Applications*, *35*(23), 17095–17112. doi:10.1007/s00521-023-08568-z

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D. Nonlinear Phenomena*, *404*, 132306. doi:10.1016/j.physd.2019.132306

Shuang, S., Qi, L., En, C., Han, W., Zhen, H., Wei, Z., Yu, S., Hai, M., & Shi, W. (2020). Convolutional knowledge tracing: Modeling individualization in student learning process. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20).* Association for Computing Machinery.

Song, X., Li, J., Cai, T., Yang, S., Yang, T., & Liu, C. (2022). A survey on deep learning based knowledge tracing. *Knowledge-Based Systems*, *258*, 110036. doi:10.1016/j.knosys.2022.110036

Sun, X., Zhao, X., Ma, Y., Yuan, X., He, F., & Feng, J. (2019). Muti-behavior features based knowledge tracking using decision tree improved DKVMN. In *Proceedings of the ACM Turing Celebration Conference*-China. Association for Computing Machinery. doi:10.1145/3321408.3322847

Wan, X., Xu, K., Liao, X., Jin, Y., Chen, K., & Jin, X. (2023). Scalable and efficient full-graph GNN training for large graphs. *Proceedings of the ACM on Management of Data*, *1*(2), 1–23. doi:10.1145/3589288

Xiong, X., Zhao, S., Van Inwegen, E. G., & Beck, J. E. (2016). *Going deeper with deep knowledge tracing* [Paper presentation]. At the International Conference on Educational Data Mining, Raleigh, North Carolina.

Yang, Y., Shen, J., Qu, Y., Liu, Y., Wang, K., Zhu, Y., Zhang, W., & Yu, Y. (2021). GIKT: a graph-based interaction model for knowledge tracing. In *Proceedings of Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD.* Springer. doi:10.1007/978-3-030-67658-2_18

Yeung, C. K., & Yeung, D. Y. (2018). Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale.* Association for Computing Machinery. doi:10.1145/3231644.3231647

Zeng, Y., Jin, T., & Dou, L. (2022, 15-18 Dec.). 2022). Difficulty-aware Convolutional Knowledge Tracing for Student Performance Prediction. In *Proceedings of 2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta)*. IEEE.

*Hu Ping, Lecturer, Master`s Degree. Graduated from Hunan Normal University in 2012. Worked in Henan Institute of Science and Technology. Her research interests include information technology education, computer supported collaborative learning and knowledge generation.*

*Li Zhaofeng, Associate professor, Master's degree. Graduated from East China Normal University In 2008. Worked in Henan Institute of Science and Technology. His research interests include artificial intelligence, virtual reality technology and digital image processing.*

*Zhang Pei, Lecturer, Master's degree. Graduated from Jilin University in 2011. Worked in Henan Institute of Science and Technology. Her research interests include digital media, virtual reality and resource development.*

*Gao Jimei, Associate professor, Master's degree. Graduated from Hunan Normal University in 2011. worked in Huanghe Jiaotong University. Her research interests include information technology education and data mining.*

*Zhang Liwei, Lecturer, Master's degree. Graduated from Zhengzhou University in 2014. Worked in Henan Institute of Science and Technology. His research interests include embedded systems, learning analytics and image processing.*