Integrating Multiplex Heterogeneous Network for Knowledge Tracing

Liqing Qiu
School of Computer Science and Engineering
Shandong University of Science and Technology, China
sun_siao@163.com

Si'ao Sun
School of Computer Science and Engineering
Shandong University of Science and Technology, China
202283060051@sdust.edu.cn

Abstract: *Knowledge Tracing (KT) predicts the probability of a student answering the subsequent question correctly based on their past performance, providing an assessment of the mastery of underlying concepts. However, the sparsity of interaction data in KT poses challenges, leading most models to represent questions using concepts and overlooking specific question information. Although the existing graph structure between concepts and questions considers both questions and concepts, current methods predominantly focus on homogeneous or heterogeneous graphs, presupposing a singular edge type linking nodes and neglecting the diverse feature paths within a multiplex heterogeneous network. Addressing these obstacles, the present study presents a model known as Integrating Multiplex Heterogeneous Network for Knowledge Tracing (MHNKT). Treating students and questions as two distinct types of nodes, they are connected through incorrect and correct interactions to construct a multiplex heterogeneous network for student-question. Employing multiplex Heterogeneous Graph Neural Networks (HetGNN), the model learns question representations from heterogeneous node aggregation and multi-type edge aggregation. Additionally, to address the many-to-many relationships between questions and concepts, the question-concept graph is input into a two-layer Graph Convolutional Network (GCN) for acquiring question representations. Results from experiments on four real datasets indicate that the MHNKT model outperforms the baseline model in terms of performance*.

Keywords: *Knowledge tracing, multiplex heterogeneous networks, graph representation learning, pre-trained embeddings, graph convolutional networks.*

Received July 22, 2024; accepted November 14, 2024
https://doi.org/10.34028/iajit/22/2/3

1. Introduction

Online education platforms, exemplified by Massive Open Online Courses (MOOCs), offer a diverse range of courses and exercises [8]. Students can plan their learning independently within these systems. Yet, without personalized guidance, students often struggle to identify their weaknesses and may not make significant progress. For example, in online learning systems, students frequently encounter challenges in managing their learning paths due to the vast amount of content and lack of real-time feedback on their knowledge gaps. This lack of guidance often leads to lower course completion rates and disengagement. Therefore, there is a critical need for Knowledge Tracing (KT) to help students track their understanding of different knowledge points (i.e., their knowledge state) and provide targeted reinforcement where necessary. By predicting the probability of correctly answering future questions based on past performance [2], KT offers personalized insights, enabling effective interventions. This, in turn, supports downstream tasks such as course recommendations, question suggestions, and knowledge structure diagnostics, making online education platforms more adaptive to individual learning needs.

Traditional KT models use the question response interaction sequence $S=(s_1, s_2,\ldots, s_t)$ of students as input. Here, $s_t=(q_t, r_t)$, where q_t represents the question ID that the student responded at time t, and r_t denotes the correctness of the student's response at time t. When $r_t=0$, the response is incorrect, and when $r_t=1$, the response is correct. In Intelligent Tutoring Systems (ITS), a significant number of questions are assigned to a small number of students, leading to data sparsity issues. To address this problem, some KT models utilize concept-indexed questions. For instance, Deep Knowledge Tracing (DKT) [16] utilizes student interaction tuples as inputs to the Recurrent Neural Network (RNN) [20]. The input vector is configured as a one-hot representation of the student $s_t=(q_t, r_t)$ for datasets with several unique questions. Self-Attentive Knowledge Tracing (SAKT) [14] utilizes student interaction tuples as inputs to compute attention weights. The interaction tuple $s_t=(q_t, r_t)$ is represented in the model as a scalar value $d_t=q_t+r_t\times E$, where E denotes the total number of questions.

Representing questions using concepts may lead to only representing concepts or questions, neglecting other available information. In KT, both questions and

knowledge concepts inherently possess graph structure features. Therefore, some researches have investigated employing Graph Neural Networks (GNNs) to improve the effectiveness of Knowledge Tracing models that do not incorporate graph structures. Nakagawa *et al*. [13] first applied GNNs to KT, proposing Graph-based Knowledge Tracing (GKT). GKT converts the knowledge structure into a graph format, reframing the KT task within GNNs as a node-level classification problem over time series data. Compared to models that only take interaction sequences as input, GKT considers relationships between concepts, thereby improving prediction accuracy.

However, the knowledge concept graph is homogeneous, only taking knowledge concepts as input and neglecting issues such as the impact between concepts and questions. Liu et al. proposed an improvement to KT through Pre-trained Question Embeddings (PEBG) [10]. This innovation represents the pioneering use of a heterogeneous graph connecting questions and knowledge concepts. PEBG extracts and leverages explicit question-concept relationships, implicit question similarity, and concept similarity information from the question-knowledge concept graph. Subsequently, its pre-trains question embeddings with this heterogeneous graph to tackle the problem of sparse interaction data.

Existing heterogeneous graphs, such as question-knowledge concept graphs, typically presuppose a single type of edge connecting two nodes. This overlooks the multi-type features existing between nodes in a multiplex heterogeneous network and fails to consider the varying significance of multiplex structures connecting nodes. To tackle this issue, the present study introduces a DKT model founded on multiplex heterogeneous GNNs. Treating a student's correct and incorrect responses as distinct behaviors, a student entity and a question entity are connected through these two behaviors, forming a multiplex heterogeneous graph for student-question interactions. The multi-type features provide a more nuanced understanding of student performance compared to single-type edges. They offer a more detailed representation, as single-type edges treat all interactions equally. To capture information from nodes in the student-question graph, a heterogeneous node aggregation network is designed. Additionally, to learn information from different types of edges, a multi-type edge aggregation network is introduced that aggregates features from different behaviors by assessing the actions' similarity across heterogeneous nodes. The multiplex heterogeneous GNN compensates for the lack of multi-type features found in standard neural networks. By integrating the multiplex heterogeneous network with the straightforward yet efficient DKT model, we validate its effectiveness while also addressing the challenge of sparse interaction data present in DKT. Inspired by the question-concept graph, this paper employs Graph Convolutional Networks (GCNs) to understand the relationships between questions and knowledge concepts. The first layer of GCN considers the direct connections between question and concept nodes, while the second layer of GCN accounts for the many-to-many relationships between questions and concepts, where a concept corresponds to multiple questions, and a question involves multiple concepts.

The main contributions of our work are summarized as follows:

1. Constructing a student-question graph, it is input into a multiplex heterogeneous GNN for learning graph representations. Node representations are learned through a heterogeneous node aggregation network and a multi-type edge aggregation network.
2. Utilizing a dual-layer GCN, we extract high-order semantic information from the graph representing questions and knowledge concepts. This approach addresses the complex associations between questions and concepts in a many-to-many manner, while alleviating the challenges posed by sparse interaction data.
3. Comprehensive experiments were performed on four real-world datasets, demonstrating that the proposed Multiplex Heterogeneous Network for Knowledge Tracing (MHNKT) model outperforms the baseline methods.

2. Related Work

2.1. Knowledge Tracing

Currently, DKT models can be categorized into two types based on input: Models that take interaction sequences as input and models that take graph embeddings as input.

DKT stands out as the pioneer in applying deep learning techniques to KT. It utilizes RNN to characterize student learning, with the RNN's hidden state serving as the representation of the student's knowledge state. DKT transforms the interaction sequence into a sequence of fixed-length vectors using either one-hot encoding or compressive sensing. This vector is then input into the RNN to obtain the student's knowledge state. The knowledge state is further processed through a linear layer with sigmoid activation, yielding the prediction results $\tilde{p} = \{p_1, p_2, \ldots, p_n\}$. Here, p_n signifies the probability of the student providing a correct response to the question. Due to sparse interaction data, the frequency of concept-response pairs is higher than that of question-response pairs. DKT addresses this by using concepts to represent questions. Additionally, the reliance on RNN in DKT introduces challenges related to long-term dependencies, limiting its ability to leverage extended input sequences.

Context-aware Attentive Knowledge Tracing (AKT) [5], proposed by Ghosh et al., takes the interaction

sequence as input but goes beyond a simple representation of questions using concepts. AKT leverages the Rasch model for generating question embeddings and question-response embeddings, considering the distinct characteristics among questions associated with a common concept. Through an enhanced monotonic attention mechanism, AKT constructs context-aware representations for questions and responses. The monotonic attention mechanism incorporates an influence factor measured by relative distance during weight calculation. As the distance increases, this influence factor gradually diminishes, resulting in less attention to distant elements. The monotonic attention mechanism not only addresses the issue of long-term dependencies but also simulates the forgetting process in students.

The GKT model, introduced by Nakagawa *et al*. [13], marked the pioneering incorporation of GNNs into the realm of KT. A graph $G=\{V, \varepsilon\}$ is formed to represent the connections among knowledge concepts, with concepts represented as $V=\{v_1, v_2, \ldots, v_n\}$, and the edge set represented as $\varepsilon \subseteq V*V$. When a student solves a question, the knowledge state $h_t=\{h_t^{i\in V}\}$ of the corresponding concept v_i changes, and the knowledge states $h_t=\{h_t^{j\in N_i}\}$ of related concepts also change, where N_i represents a group of nodes adjacent to v_i. The concept graph used by GKT is homogenous, considering only relationships between concepts while ignoring other information.

A Graph-based Interaction Model for Knowledge Tracing (GIKT), proposed by Yang *et al*. [23], constructs a question-concept graph based on the many-to-many relationships between questions and concepts. A concept may correspond to multiple questions, and a question may cover different concepts. The question-concept graph contains nodes of two distinct types, forming a heterogeneous graph. GIKT employs GCNs for extracting high-order relational information from the question-concept graph, using the learned representations as question embeddings. These embeddings for questions, coupled with corresponding embeddings for answers, function as inputs for the KT model. Additionally, GIKT includes a reformulation module and an interaction module, offering a unified approach to more effectively model students' understanding of new questions and their associated knowledge concepts. GIKT alleviates data sparsity issues and considers relationships between questions and concepts.

In the continuous advancement of KT, graph embeddings not only effectively address the issue of sparse interaction data but also offer a more structured approach to represent the complex relationships between students, questions, and knowledge concepts. Our goal is to extract richer information from these graphs, further enhancing the understanding and prediction of student learning processes. To this end, this paper proposes two distinct graph structures: A question-concept graph and a student-question graph. These graphs enable us to capture the connections between questions and knowledge points, while also analyzing student performance across different questions, thus providing more accurate support for personalized learning paths.

2.2. Graph Neural Network

The aim of GCN is to extract node representations through convolution operations on graphs, enhancing the effectiveness of subsequent tasks.

GCNs operate on homogeneous graphs, wherein all nodes and edges share the same type. Yet, they suffer from issues like over-smoothing, especially in deep architecture. In response to these challenges, methods like inductive representation learning on large graphs GraphSAGE [6] have been introduced. Graph-SAmple and aggreGatE (GraphSAGE) is designed for generating node embeddings through the sampling and aggregation of information from the neighborhood, making it suitable for scalable and efficient representation learning on large graphs. Another approach, Graph Attention Networks (GAT) [18], introduces the attention mechanism to assign different importance weights to neighbor nodes during aggregation. This enhances the expressiveness of node representations by allowing nodes to attend to their neighbors selectively.

A major limitation of homologous neural networks is to process graphs where nodes and edges share the same type. In a real-world scenario, a graph may contain multiple types of nodes and edges, such as students, problems, knowledge points, and so on. To capture these complex relationships better, a heterogeneous GNNs is introduced. Heterogeneous Graph Attention Network (HAN) [19] is designed for heterogeneous graphs, allowing for nodes and edges to be affiliated with distinct types. It incorporates an attention mechanism to assign varying weights to different node types during message aggregation. While not a GNN, Metapath2Vec [3] serves as a technique for learning representations in heterogeneous graphs. It utilizes meta-paths, which are sequences of node types and edge types, to generate embeddings for nodes. Heterogeneous Graph Neural Network (HetGNN) [25] serves as a general framework for learning in heterogeneous graphs. It employs different convolutional operations for different types of nodes and edges, allowing flexibility in capturing various relationships.

Although heterogeneous GCNs have addressed the limitations of homogeneous GCNs on heterogeneous graphs, they often assume that each type of relation in the graph has only one type of edge, overlooking the multiplex features between different types of nodes. Recently, numerous multiplex network embedding techniques have been proposed. Fast Attributed

Multiplex Heterogeneous Network Embedding (FAME) [11] achieves efficient embedding learning for network topology, node attributes, and relationships through scalable spectral transformation and sparse random projection. This approach automatically preserves attribute semantics and multiple types of relations (multiplex network) in the learned embeddings. Heterogeneous Graph Structure Learning for graph neural networks (HGSL) [27] considers the heterogeneity of different relations by generating each relation subgraph, including feature similarity graph, feature propagation graph, and semantic graph. The integration of these subgraphs forms a learned heterogeneous graph, and together with a GNN, it is optimized for classification objectives. This approach contributes to better adaptation to the features and semantics of different relations in heterogeneous graphs.

Inspired by the aforementioned GNNs, we treat a student's correct and incorrect responses as two distinct behaviors and represent them as different edges connecting student and question nodes. The resulting graph is referred to as the student-question graph, where the different types of edges are termed multiplex features. The student-question graph is then used as input to a multiplex heterogeneous graph network to learn question representations. To address the issue of sparse interaction data, we use a question-concept graph to represent the relationships between questions and knowledge concepts, rather than simply indexing questions by concepts. Through a two-layer GCN, we capture high-level semantic relationships between questions and concepts.

3. Preliminaries

This section formally defines the problem studied and introduces the key symbols used in the paper. Typically, a network is represented as $G=\{V, \varepsilon\}$, where V comprises nodes, and ε comprises edges connecting the nodes, signifying relationships between them. Linked with a node type mapping function $\phi: V \rightarrow O$ and an edge type mapping function $\psi: \varepsilon \rightarrow R$, where O signifies the set of all node types and R denotes the set of all edge types.

3.1. Multiplex Heterogeneous Graph

A multiplex heterogeneous network, denoted as $G=\{V, \varepsilon, X\}=\{G_1, G_2, \ldots, G_{|R|}\}$, consists of $G_r=\{V, \varepsilon_r, X\}$, a subgraph with edge type $r \in R$. Here, $V=\cup_{o \in O} V_o$, $\varepsilon=\cup_{r \in R} \varepsilon_r$, and $X \in R^{n \times m}$. X is the node feature matrix, where n represents the size of the node set V, and m denotes the size of attribute features. The construction of the node feature matrix X is not fixed. For $|O|+|R|>2$ and diverse edge types exist between the identical pairs of nodes, the network is termed multiplex heterogeneous.

3.2. Student-Question Graph

As two different types of nodes, students and questions are connected by two different behaviors: Wrong response and right response. According to section 3.1, the student question graph is a multiplex heterogeneous graph. G^{uq} represents the student question graph and A denotes the adjacency matrix corresponding to this graph.

3.3. Question-Concept Graph

Every question q_i is associated with one or more concepts, and each concept c_i usually corresponds to more than one question. Questions and concepts act as two different types of nodes, connected by correspondence between questions and concepts. The obtained question concept diagram is a heterogeneous network. G^{qc} represents the question concept graph and C represents the adjacency matrix of the question concept graph.

4. Method

In this section, a detailed overview of the MHNKT framework will be presented, showcasing the overall architecture depicted in Figure 1. The MHNKT model is composed of three components: multiplex HetGNN, multi-layer GCNs, and DKT. To provide better clarity, Table 1 presents a summary of the notations used in this study.

Table 1. Notations.

Notation	Description
$S=(s_1, s_2, \ldots\ldots, s_t)$	The historical interaction records of students
s_t	Student interaction
q_t	Question ID.
r_t	It indicates whether the student's response is correct.
V and ε	Node set and edge set
X	Node feature matrix of node
G_r	It is a subgraph with edges type
G^{uq} and A	Student-question graph and its adjacency matrix
G^{qc} and C	Question-concept graph and its adjacency matrix
A_{node} and H_{node}	Aggregation matrix and node representation of heterogeneous node aggregation module
A_{edge} and H_{edge}	Similarity matrix and node representation of multi-type edge aggregation module
x_t	Question embedding
y_t	Forecasted value

4.1. Multiplex Heterogeneous Graph

The primary difference between general heterogeneous networks and multiplex heterogeneous networks lies in the presence of multiplex structures among nodes, granting the network multiplex distinct perspectives. Consequently, considering this multiplex structure becomes crucial for the embedding of multiplex heterogeneous networks. The multiplex structures involve more extensive semantic information, including aspects like mutual promotion or inhibition.

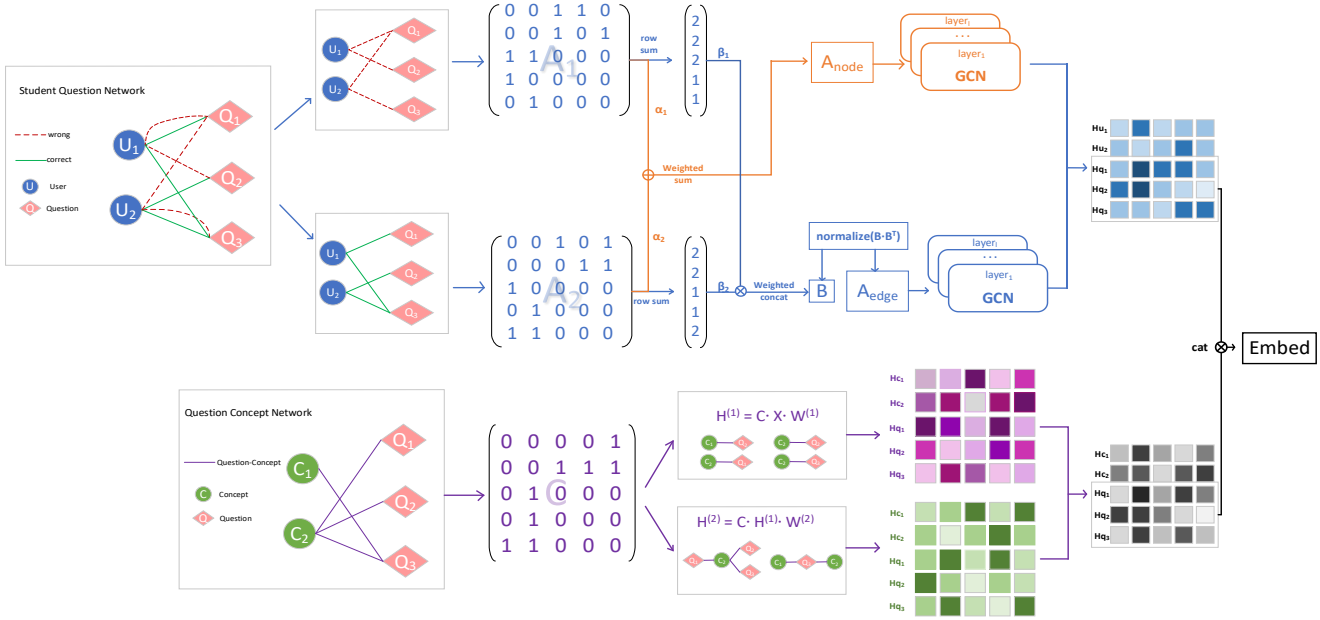

Figure 1. The overview of MHNKT. The upper part of the figure illustrates a multiplex HetGNN, while the lower part depicts a multi-layer GCN.

4.1.1. Heterogeneous Node Aggregation

In a multiplex heterogeneous network, each type of edge plays a distinct role and has different influences on node representations. Consequently, the first step is to decouple the multiplex heterogeneous network based on the types of edges [17]. The generated subgraphs are represented as $\{G_r|r=1, 2, \ldots, |R|\}$, and their corresponding adjacency matrices are $\{A_r \in R^{n\times n}|r=1, 2, \ldots, |R|\}$. In the student-question network, there are two subgraphs. The graph connected by the incorrect response behavior is denoted as G_1^{uq} with an adjacency matrix A_1. The student-question network connected by the correct response is denoted as G_2^{uq} with an adjacency matrix A_2.

The diverse behaviors of students (incorrect response, correct response) reflect different preferences in addressing questions. Consequently, multiple student-question interactions with distinct semantic relationships have varying impacts on the learning process of question representations. To capture these diverse types of node dependencies, the heterogeneous node aggregation module aggregates node features from multiple structures [24]. This module initially employs a set of trainable weight parameters to aggregate different subgraphs:

$$A_{node} = \sum_{r=1}^{|R|} \alpha_r A_r \tag{1}$$

where R means there are several types of edges.

Then the aggregation matrix $A_{node} \in R^{n\times n}$ is imported into the convolutional network for convolution operations to extract the relationships in the student problem graph.

$$H_{node}^{(1)} = A_{node} \cdot X \cdot W_{node}^{(1)} \tag{2}$$

The resulting $H_{node}^{(1)}$ is the node representation learned by single-layer GCN. Where $X \in R^{n\times m}$ is a node feature matrix, $W^{(1)} \in R^{m\times d}$ is a learnable weight.

To comprehend deeper network features, we have the option to extend it to the L-layer:

$$H_{node}^{(l)} = A_{node} \cdot H_{node}^{(l-1)} \cdot W_{node}^{(l)} = A_{node}^{(l)} \cdot X \cdot W_{node}^{(1)} \cdots W_{node}^{(l)} \tag{3}$$

The depth of the heterogeneous node aggregation module is determined by the count of convolution layers.

Finally, the output from all layers is fused to acquire the node representation learned from the heterogeneous node aggregation module.

$$H_{node} = \frac{1}{l} \sum_{i=1}^{l} H_{node}^{(i)} \tag{4}$$

4.1.2. Multi-Type Edge Aggregation

The embedding matrices of the two subgraphs in the student-question graph contain only relevant information for each relationship type, thus failing to leverage the multiplicity of the network. To address this, a multi-type edge aggregation module is designed to characterize the multiple behaviors between nodes from the perspective of edge types and quantities. In the student-question network, the behavior of students answering similar questions tends to be more similar. In multi-type edge aggregation, the connection strength between two nodes is dictated by the similarity of edges. This not only reveals the strength of connections between students and questions but also indicates the similarity between different questions. Similar questions are likely to involve the same concepts, leading to more similar behaviors for questions with identical concepts.

In this module, the matrices representing the two

subgraphs are summed row-wise, resulting in two-column vectors. Each column vector describes the quantity of corresponding behaviors for each node relative to all other nodes. Using a set of learnable weight values β_i, the column vectors specific to edge types are concatenated, producing the multi-type edge matrix $B \in R^{n \times R}$.

$$B_r[j] = \sum_{k=1}^{n} A_r[j,k] \tag{5}$$

$$B = \beta_1 \cdot B_1 \parallel \beta_2 \cdot B_2 \parallel \cdots \parallel \beta_R \cdot B_R = (B_1 \parallel \cdots \parallel B_R) \cdot \Lambda_\beta \tag{6}$$

$B_r \in R^{n \times 1}$ is a column vector, and each column vector corresponds to an edge of one type. The symbol $\parallel$ represents the concatenation operation, and Λ_β is a trainable diagonal matrix defined as $diag(\beta_1, \beta_2, \ldots, \beta_R)$.

Then, perform the multiplication of the multi-type edge matrix by its transpose and normalize the outcome to derive the multi-type edge similarity matrix.

$$A_{edge} = normalize(B \cdot B^T) \tag{7}$$

Obtain $A \in R^{n \times n}$, which is the matrix representing the similarity degrees of edges.

Next, input the matrix A_{edge} obtained above into the GCN to aggregate information, resulting in node representations based on the similarity of multi-type edges.

$$H_{edge}^{(1)} = A_{edge} \cdot X \cdot W_{edge}^{(1)} \tag{8}$$

To comprehend deeper network features, we have the option to extend it to the L-layer:

$$H_{edge}^{(l)} = A_{edge} \cdot H_{edge}^{(l-1)} \cdot W_{edge}^{(l)} = A_{edge}^{(l)} \cdot X \cdot W_{edge}^{(1)} \cdots W_{edge}^{(l)} \tag{9}$$

The obtained $H_{edge}^{(l)}$ is the node representation learned by the last layer of GCN. Note that the paper sets the count of convolutional layers as 2.

The multi-type edge aggregation module indicates that, despite being distant in the network topology, their learned representation vectors will be relatively correlated if their edge similarities are significant. Furthermore, there is a close similarity in the probability of a student answering similar questions correctly.

Finally, average pooling is employed on the results from both the heterogeneous node aggregation module and the multi-type edge aggregation module to obtain the ultimate node representation $H \in R^{n \times d}$, where:

$$H = \frac{1}{2}(H_{node} + H_{edge}) \tag{10}$$

4.2. Multi-Layer Graph Convolutional Networks

To tackle the problem of sparse interaction data, this model leverages the question-concept network to extract the relationships between questions and concepts, rather than simply representing questions using concepts. The question- concept graph G^{qc} is a heterogeneous graph with its adjacency matrix denoted as $C \in R^{n \times n}$. In this model, a multi-layer GCN is employed to learn the representation of the question-concept graph [24]. The learned node representation encompasses high-order semantic relationships between questions and concepts. The immediate neighbors of a question should correspond to its associated concepts, while the secondary neighbors should consist of other questions sharing the same concepts [23].

A multilayer GCN is applied to process the question-concept graph. The first GCN layer considers the direct connections between the problem and concept nodes:

$$H^{(1)} = C \cdot X \cdot W^{(1)} \tag{11}$$

where $H^{(1)} \in R^{n \times d}$ represents the output of the first layer, $X \in R^{n \times m}$ is the node feature matrix, and $W^{(1)} \in R^{m \times d}$ is the trainable weight matrix. By capturing these immediate relationships, the first layer helps to establish a foundational understanding of how questions relate to specific concepts, addressing the sparsity of interaction data by connecting each question to its relevant concepts.

The second GCN layer considers the many-to-many relationships between question and concept nodes, where one concept corresponds to multiple questions, and one question includes multiple concepts. Thus, the count of convolutional layers in the multi-layer GCN is designated as 2:

$$H^{(2)} = C \cdot H^{(1)} \cdot W^{(2)} \tag{12}$$

where $W^{(2)} \in R^{d \times d}$ represents the trainable weight matrix for the second layer. This layer helps to capture more complex interactions and shared knowledge between questions, enhancing the representation by aggregating information from a broader context.

The choice of a two-layer GCN is motivated by the need to balance model complexity with the richness of the information extracted [9]. While deeper networks can capture more intricate patterns, they may also introduce noise or overfitting, especially in the presence of sparse data. Therefore, a two-layer configuration is optimal for capturing the necessary semantic relationships without sacrificing generalization.

Finally, the outputs of the single-layer GCN and the double-layer GCN are fused:

$$H = \frac{1}{2}(H^{(1)} + H^{(2)}) \tag{13}$$

where H is the final node representation learned through the multi-layer GCN.

4.3. Deep Knowledge Tracing

DKT is the prediction module of the MHNKT model.

The paper employs a DKT model based on Long Short-Term Memory (LSTM) for this purpose [17, 23]. The final representation, obtained by concatenating the learned representations of the question nodes from the multiplex heterogeneous network and the multi-layer convolutional network, is utilized as pre-trained input. This pre-trained representation is fed into an LSTM, where the LSTM's hidden state is regarded as the student's knowledge state. Based on the student's knowledge state, the model forecasts the likelihood of the student answering the next question correctly. Equations (14) to (20) illustrate the process of LSTM modeling the student's learning situation. Let h_t be the hidden state at time step t, x_t be the input at time step t (in this case, the PEBG), and f_t, i_t, o_t be the forget gate, input gate, and output gate vectors, respectively.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{14}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{15}$$

$$\tilde{c}t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{16}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}t \tag{17}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{18}$$

$$h_t = o_t \odot \tanh(c_t) \tag{19}$$

$$y_t = \sigma(W_y h_t + b_y) \tag{20}$$

where σ is the sigmoid activation function, $\odot$ denotes element-wise multiplication. W_f, W_i, W_c, W_o, and W_y are weight matrices for the forget gate, input gate, cell state, output gate, and predict, respectively. b_f, b_i, b_c, b_o, b_y are bias vectors. $[h_{t\text{-}1}, x_t]$ denotes the concatenation of $h_{(t\text{-}1)}$ and x_t.

The MHNKT model undergoes training with the cross-entropy loss, quantifying the disparity between the observed response r_t and the predicted value y_t. The objective is to minimize this loss, and the optimization is carried out through the Adam optimizer.

$$l = -\sum_t (r_t \log y_t + (1 - r_t) \log(1 - y_t)) \tag{21}$$

5. Experiment

This section presents an extensive series of experiments conducted on four real-world datasets to evaluate the performance of the MHNKT model. The superior performance of MHNKT is proved by comparative experiments. Additionally, ablation experiments are carried out to offer empirical support for the effectiveness of the proposed enhancements in the MHNKT model.

5.1. Datasets

Four publicly accessible datasets are employed in this study to assess the performance of various KT models. While these four datasets vary in scope and context, they collectively represent a diverse range of educational settings, from blended learning environments to traditional classroom settings. Their selection was based on their availability, size, and relevance to current KT research. By utilizing datasets from different educational contexts, we aim to ensure that our findings are more generalizable and reflective of real-world educational challenges, providing a comprehensive evaluation of our proposed model's effectiveness. Table 2 summarizes general information about each dataset. The detailed information is as follows:

1. Assist2009[1]: This dataset consists of 123 concepts, 17751 questions, and 4163 users. Known as the skill builder dataset or Mastery Learning dataset, it defines mastery as students continuously answering three questions correctly. Once mastered, no further questions are recommended for that skill, making it an effective measure of learning progress [4].
2. Assist2017[2]: Comprising 102 concepts, 3,162 questions, and 1,709 users, this dataset was obtained from a longitudinal study that tracked students' usage of the ASSISTments blended learning platform during their middle school years from 2004 to 2007. Its use in the 2017 ASSISTments data mining competition underscores its relevance and robustness for evaluating educational models [15].
3. Statics2011[3]: This dataset features 1,179 concepts, 300 questions, and 335 users, representing student work from an engineering statics course at Carnegie Mellon University in fall 2011. It connects problem names and steps as knowledge concepts, offering a unique perspective on engineering education that is often underrepresented in other datasets [7].
4. Junyi[4]: Considering the large size of the entire dataset, 5000 students were randomly selected in the JUNYI dataset. The processed dataset included 1171 concepts, 703 questions, and 5,000 students. The dataset comprises logs of problems and details related to exercises from Junyi Academy, an online learning platform initiated in 2012, built upon the open-source code provided by Khan Academy [1].

Table 2. The key characteristics of the datasets.

Characteristic	Assist2009	Assist2017	Statics2011	Junyi
Users	4163	1709	335	5000
Concepts	123	102	1179	1171
Questions	17751	3162	300	703
Records	401757	942817	361092	5000

[1]https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data/skill-builder-data-2009-2010
[2]https://sites.google.com/view/assistmentsdatamining/data-mining-competition-2017
[3]https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=507
[4]https://pslcdatashop.web.cmu.edu/Files?datasetId=1198

5.2. Baselines

1. DKT [16] investigates the effectiveness of employing RNNs for modeling student learning. By leveraging neural networks, significant enhancements in prediction performance have been observed across various KT datasets.
2. Dynamic Key-Value Memory Networks (DKVMN) [26] leverage the connections among fundamental concepts to directly track a student's proficiency level in each concept. DKVMN comprises a fixed matrix referred to as the "key" used to retain knowledge concepts, and a flexible matrix named the "value" responsible for preserving and revising the proficiency levels of the associated concepts.
3. GKT [13] constructs a knowledge concept map that learns the representation of nodes through GNN. Refactoring KT tasks into GNN Node-level classification of time series can improve prediction accuracy without any additional information.
4. A GIKT [23] constructs a question-concept graph that aggregates question embeddings and concept embeddings through GCN. Questions and concepts represent diverse manifestations of knowledge, GIKT extends the evaluation of a student's proficiency in a question to encompass interactions involving the student's present state, historical states, the target question, and associated concepts.
5. Difficulty and Attempts boosted Graph-based Knowledge Tracing (DAGKT) [12] also employs a question-concept graph to aggregate relationships between questions and concepts. DAGKT suggests that questions sharing the same knowledge concept may exhibit varying levels of difficulty, and different attempts by students represent distinct levels of knowledge mastery. The model integrates question difficulty and student attempts into embeddings for both questions and answers.
6. Knowledge Tracing based on a Heterogeneous Information Network (HINKT) [22] obtains implicit relationships among questions or concepts through an examination of the interactions between students and the entities of questions and concepts. It marks the first KT model to incorporate three entities simultaneously in constructing a heterogeneous information network.
7. The Dual-channel Heterogeneous Graph Network (DHGN) [21] learns informative representations of questions by leveraging both meta-path and network pattern channels to capture high-order and local relations. To enhance heterogeneous graph modeling, DHGN incorporates a self-supervised task.

5.3. Implementation Details

In the proposed MHNKT, the dimensions of all pre-trained embeddings are set to 200. The LSTM hidden layer has a dimension of 200. Both the convolutional layers in the multiplex heterogeneous network and the multi-layer GCN are set to 2 layers to avoid the issue of excessive smoothing. ReLU activation is used in the convolutional network. For each dataset, 80% of the sequences constitute the training set, while the remaining 20% form the test set. To prevent overfitting during training, an early stopping mechanism is employed, where the model stops training if there is no improvement in performance for consecutive 20 rounds. Adam's optimization algorithm is utilized for optimizing all trainable parameters. Other hyperparameters, including dropout in GCN, learning rate, and batch size during training, will be determined through parameter-tuning discussions. The evaluation metric utilized by the model is the Area Under the ROC Curve (AUC).

5.4. Hyperparameter Experiment of MHNKT

The MHNKT model incorporates several hyperparameters that are crucial in determining the model's prediction performance. These hyperparameters include the batch size used during training, the dropout rate, and the learning rate. This section conducts a sequence of experiments to examine how these hyperparameters impact the predictive performance of the model.

5.4.1. Dropout Rate

In this section, the experiments are carried out to determine the appropriate dropout rate. The experiments are conducted on four real datasets, with the batch_size=128, and a learning rate ξ=1e-5. The dropout rate varied within the range [0.1, 0.5]. Figure 2 illustrates the variation of AUC with different dropout rates. From Figure 2, it can be observed that the fluctuation trends of the curves are not entirely consistent. Considering the AUC values across all four datasets, when the dropout rate is set to 0.4, the model achieves relatively high AUC values on all four datasets. Therefore, a dropout rate of 0.4 is set in the model.

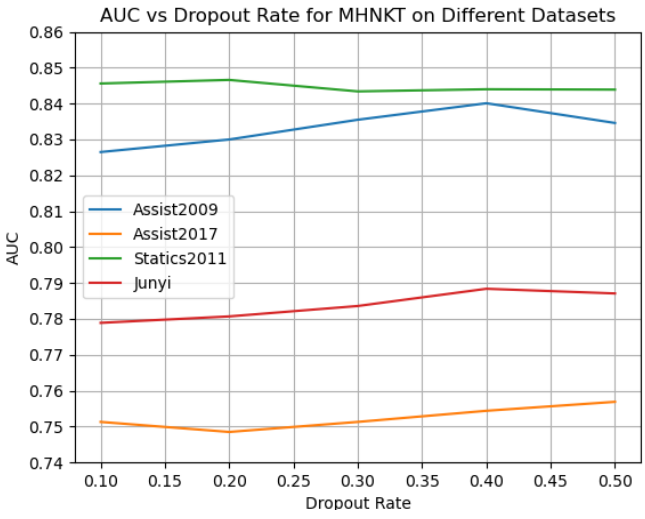

Figure 2. The AUC exhibits variations across distinct dropout rate values.

5.4.2. Learning Rate

This section provides a detailed experimental procedure for determining the initial learning rate. The experiments are carried out using four real datasets, with the batch_size=128, and dropout rate λ=0.4. The initial learning rate ξ is varied within the given range [1e-5, 0.1], with one order of magnitude 0.1 increments. In the model, the learning rate is fixed during training. Figure 3 illustrates the variation of AUC under different learning rates. When the learning rate ξ=0.001, the AUC values peak across all datasets. Therefore, the model's initial learning rate is established at 0.001.

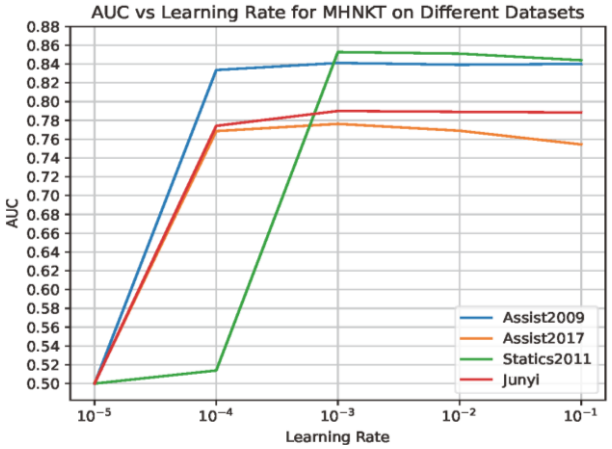

Figure 3. The AUC exhibits variations across distinct learning rate values.

5.4.3. Batch Size

In this part, the experimental procedure for determining the appropriate batch_size hyperparameter is presented in detail. The experiments are performed using four real datasets, with the dropout rate λ=0.4, and learning rate ξ=0.001. The batch_size is set to values of 2^n, where n ranges from 3 to 7 with an interval of 1. Figure 4 illustrates the correlation between the AUC and the batch_size. The outcomes indicate that when n is within the range of 5 to 7, increasing the batch_size results in a gradual decline in AUC. To account for the different sizes of the datasets, set batch_size to 32.

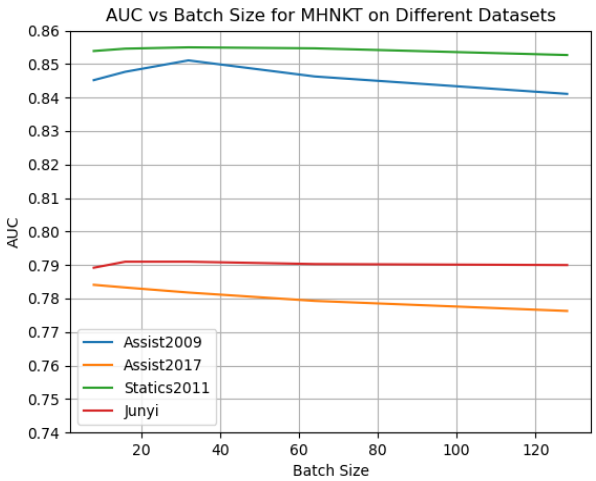

Figure 4. The AUC exhibits variations across distinct batch_size.

5.5. Multiplex Heterogeneous Graph

Table 3 presents each model's AUC results on the four datasets. As observed, MHNKT achieves the best performance across all four datasets. On the datasets assist2009, assist2017, statics2011, and Junyi, the AUC achieved by our proposed model is 2.46%, 0.93%, 2.50%, and 2.08% higher, respectively, compared to the state-of-the-art baseline model DHGN. The first two baselines are classical KT methods, the third is a homogeneous network approach, and the last four are heterogeneous network approaches, specifically:

a) The GNNs-based GKT demonstrates superior performance compared to the LSTM-based DKT, indicating the effectiveness of graph structures.
b) HetGNN-based models GIKT, DAGKT, HINKT, and DHGN outperform the use of homogeneous neural networks like GKT, highlighting the effectiveness of considering relationships between questions and concepts.
c) MHNKT outperforms KT models based on HetGNNs, demonstrating the superiority of the multiplex HetGNN.

MHNKT utilizes multiplex heterogeneous networks and multi-layer GCNs to capture relationships between different types of nodes. In the multiplex HetGNN, information between students and questions is aggregated through the heterogeneous node aggregation module, while the multi-type edge aggregation module emphasizes the different impacts of various behaviors. In the multi-layer GCN, the consideration of many-to-many relationships between questions and concepts addresses the issue of sparse interaction data.

Table 3. Performance prediction outcomes of the baseline models.

Models	Assist2009	Assist2017	Statics2011	Junyi
DKT	0.7392	0.7094	0.7419	0.7164
DKVMN	0.7706	0.7173	0.7836	0.7452
GKT	0.7630	0.7248	0.7864	0.7238
GIKT	0.7760	0.7412	0.7811	0.7463
DAGKT	0.7869	0.7498	0.7897	0.7518
HINKT	0.8193	0.7512	0.8146	0.7557
DHGN	0.8265	0.7702	0.8300	0.7702
MHNKT	**0.8511**	**0.7795**	**0.8550**	**0.7910**

5.6. Performance Comparison

In this segment, we perform an ablation study to examine the influence of individual components in the MHNKT model. We assess four versions of the MHNKT model, each excluding one element from the original configuration. The details of these versions are provided below:

- MHNKT-RMH (remove multiplex heterogeneous network) removes the multiplex HetGNN, no longer learning question representations from the student-question graph.
- MHNKT-RMC (remove multi-layer GCNs) removes the multi-layer GCN, disregarding the relationships

between questions and concepts.

- MHNKT-RH (remove heterogeneous node aggregation) removes the heterogeneous node aggregation module from the multi-path HetGNN, no longer capturing dependencies between nodes from their perspectives.
- MHNKT-RM (remove multi-type edge aggregation) removes the multi-type edge aggregation module from the multi-path HetGNN, disregarding the similarity of edges between nodes.

According to Table 4, it is evident that the proposed components of the model contribute effectively to improving AUC. After removing the multiplex HetGNN, the model exhibits the most significant performance decline (2%). This is because the MHNKT-RMH model fails to capture the impact of different behaviors between students and questions. Upon removing the multi-layer GCN, the model's performance averages a decline of 1.5%. This validates the importance of distinguishing between questions and concepts. Utilizing concept indices to address sparse interaction data resolves the issue but overlooks the relationships between questions and concepts. After removing the multiplex HetGNN, the model's performance averages a decrease of 0.63%. Heterogeneous node aggregation aggregates node feature information and learning question representations from the student-question graph and considers not only the dependencies between nodes but also implicitly includes student features. Following the removal of the multi-type edge aggregation module, the model's performance experiences an average decline of 0.64% across all four datasets. The multi-type edge aggregation module facilitates the transfer of feature information between nodes based on edge similarity, aiding in better learning of node representations. The AUC in all four ablation experiments is lower than that of the overall model, confirming the crucial role played by each module in pre-training embedding learning for KT.

Table 4. Performance prediction outcomes of the ablation study.

Models	Assist2009	Assist2017	Statics2011	Junyi
MHNKT-RMH	0.8201	0.7443	0.8518	0.7803
MHNKT-RMC	0.8156	0.7730	0.8475	0.7778
MHNKT-RH	0.8414	0.7686	0.8525	0.7891
MHNKT-RM	0.8329	0.7736	0.8546	0.7900
MHNKT	**0.8511**	**0.7795**	**0.8550**	**0.7910**

6. Conclusions

This paper proposes a KT model, MHNKT, based on a multiplex HetGNN. MHNKT consists of two key components: a multiplex HetGNN and a multi-layer GCN. The multiplex HetGNN comprises two crucial modules: heterogeneous node aggregation and multi-type edge aggregation. The heterogeneous node aggregation module learns node representations by aggregating feature information from student nodes and question nodes, focusing on the features of nodes and considering dependencies between nodes. The multi-type edge aggregation module treats edge similarity as the connection strength between nodes. The multi-layer GCN, set to two convolutional layers, considers the many-to-many relationships that exist between questions and concepts. In contrast to solving sparse interaction data by using concepts to index questions, the multi-layer GCN leverages the inherent graph structures of questions and concepts to capture their relationships more comprehensively. This approach makes better use of the information associated with questions and concepts, addressing the challenge of sparse interaction data more effectively.

Acknowledgment

This paper is supported by the Shangdong Nature Science Foundation of China (grant no. ZR2020MF044).

References

[1] Chang H., Hsu H., and Chen K., "Modeling Exercise Relationships in E-Learning: A Unified Approach," *in Proceedings of the 8th International Conference on Educational Data Mining*, Madrid, pp. 532-535, 2015. https://www.educationaldatamining.org/EDM2015/proceedings/short532-535.pdf

[2] Corbett A. and Anderson J., "Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge," *User Modeling and User-Adapted Interaction*, vol. 4, pp. 253-278, 1994. https://link.springer.com/article/10.1007/BF01099821

[3] Dong Y., Chawla N., and Swami A., "Metapath2vec: Scalable Representation Learning for Heterogeneous Networks," *in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Halifax, pp. 135-144, 2017. https://doi.org/10.1145/3097983.3098036

[4] Feng M., Heffernan N., and Koedinger K., "Addressing the Assessment Challenge with an Online System that Tutors as it Assesses," *User Modeling and User-Adapted Interaction*, vol. 19, pp. 243-266, 2009. https://link.springer.com/article/10.1007/s11257-009-9063-7

[5] Ghosh A., Heffernan N., and Lan A., "Context-Aware Attentive Knowledge Tracing," *in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, California, pp. 2330-2339, 2020. https://doi.org/10.1145/3394486.3403282

[6] Hamilton W., Ying Z., and Leskovec J., "Inductive

Representation Learning on Large Graphs," *in Proceedings of the 31st International Conference on Neural Information Processing Systems*, California, pp. 1025-1035, 2017. https://dl.acm.org/doi/10.5555/3294771.3294869

[7] Koedinger K., Baker R., Cunningham K., Skogsholm A., Leber B., and Stamper J., *Handbook of Educational Data Mining*, CRC Press, 2010. DOI:10.1201/b10274-6

[8] Le C., Pardos Z., Meyer S., and Thorp R., "Communication at Scale in a MOOC Using Predictive Engagement Analytics," *in Proceedings of the 19th International Conference on Artificial Intelligence in Education*, London, pp. 239-252, 2018. https://link.springer.com/chapter/10.1007/978-3-319-93843-1_18

[9] Li G. and Wang F., "Image Object and Scene Recognition Based on Improved Convolutional Neural Network," *The International Arab Journal of Information Technology*, vol. 21, no. 5, pp. 925-937, 2024. https://doi.org/10.34028/iajit/21/5/13

[10] Liu Y., Yang Y., Chen X., Shen J., Zhang H., and Yu Y., "Improving Knowledge Tracing via Pre-Training Question Embeddings," *in Proceedings of the 29th International Joint Conference on Artificial Intelligence*, Yokohama, pp. 1577-1583, 2020. https://www.ijcai.org/proceedings/2020/0219.pdf

[11] Liu Z., Huang C., Yu Y., Fan B., and Dong J., "Fast Attributed Multiplex Heterogeneous Network Embedding," *in Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, Ireland, pp. 995-1004, 2020. https://doi.org/10.1145/3340531.3411944

[12] Luo R., Liu F., Liang W., Zhang Y., Bu C., and Hu X., "DAGKT: Difficulty and Attempts Boosted Graph-based Knowledge Tracing," *in Proceedings of the 29th International Conference on Neural Information Processing*, New Delhi, pp. 255-266, 2022. https://doi.org/10.1007/978-3-031-30108-7_22

[13] Nakagawa H., Iwasawa Y., and Matsuo Y., "Graph -based Knowledge Tracing: Modeling Student Proficiency Using Graph Neural Network," *in Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, Thessaloniki, pp. 156-163, 2019. https://doi.org/10.1145/3350546.3352513

[14] Pandey S. and Karypis G., "A Self-Attentive Model for Knowledge Tracing," *in Proceedings of the 12th International Conference on Educational Data Mining*, Montreal, pp. 384-389, 2019. https://drive.google.com/file/d/1yyfa9sWlVRHAY4JEQyRFZH0lge_p2HGA/view

[15] Patikorn T., Heffernan N., and Baker R., "Assistments Longitudinal Data Mining Competition 2017: A Preface," *in Proceedings of the Workshop on Scientific Findings from the ASSISTments Longitudinal Data Competition during the 11th Conference of Educational Data Mining*, Buffalo, pp. 1-11, 2018. https://sites.google.com/view/assistmentsdatamining/data-mining-competition-2017

[16] Piech C., Bassen J., Huang J., Ganguli S., Sahami M., Guibas L., and Sohl-Dickstein J., *Advances in Neural Information Processing Systems*, Curran Associates, 2015. https://papers.nips.cc/paper_files/paper/2015/hash/bac9162b47c56fc8a4d2a519803d51b3-Abstract.html

[17] Song X., Li J., Lei Q., Zhao W., Chen Y., and Mian A., "Bi-CLKT: Bi-Graph Contrastive Learning Based Knowledge Tracing," *Knowledge-Based Systems*, vol. 241, pp. 108274, 2022. https://doi.org/10.1016/j.knosys.2022.108274

[18] Velickovic P., Cucurull G., Casanova A., Romero A., Lio P., and Bengio Y., "Graph Attention Networks," *arXiv Preprint*, vol. arXiv:1710.10903, pp. 1-12, 2017. https://arxiv.org/abs/1710.10903

[19] Wang X., Ji H., Shi C., Wang B., Ye Y., Cui P., and Yu P., "Heterogeneous Graph Attention Network," *in Proceedings of the World Wide Web Conference*, San Francisco, pp. 2022-2032, 2019. https://doi.org/10.1145/3308558.3313562

[20] Williams R. and Zipser D., "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *Neural Computation*, vol. 1, no. 2, pp. 270-280, 1989. DOI:10.1162/neco.1989.1.2.270

[21] Wu T. and Ling Q., "Fusing Hybrid Attentive Network with Self-Supervised Dual-Channel Heterogeneous Graph for Knowledge Tracing," *Expert Systems with Applications*, vol. 225, pp. 120212, 2023. https://doi.org/10.1016/j.eswa.2023.120212

[22] Xu J., Huang X., Xiao T., and Lv P., "Improving Knowledge Tracing via a Heterogeneous Information Network Enhanced by Student Interactions," *Expert Systems with Applications*, vol. 232, pp. 120853, 2023. https://doi.org/10.1016/j.eswa.2023.120853

[23] Yang Y., Shen J., Qu Y., Liu Y., Wang K., Zhu Y., Zhang W., and Yu Y., "GIKT: A Graph-based Interaction Model for Knowledge Tracing," *in Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference*, Ghent, pp. 299-315, 2021. https://doi.org/10.1007/978-3-030-67658-2_18

[24] Yu P., Fu C., Yu Y., Huang C., Zhao Z., and Dong J., "Multiplex Heterogeneous Graph Convolutional Network," *in Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Washington (DC), pp. 2377-2387, 2022. https://doi.org/10.1145/3534678.3539482

[25] Zhang C., Song D., Huang C., Swami A., and Chawla N., "Heterogeneous Graph Neural Network," *in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Anchorage, pp. 793-803, 2019. https://doi.org/10.1145/3292500.3330961

[26] Zhang J., Shi X., King I., and Yeung D., "Dynamic Key-Value Memory Networks for Knowledge Tracing," *in Proceedings of the 26th International Conference on World Wide Web*, Perth, pp. 765-774, 2017. https://doi.org/10.1145/3038912.3052580

[27] Zhao J., Wang X., Shi C., Hu B., Song G., and Ye Y., "Heterogeneous Graph Structure Learning for Graph Neural Networks," *in Proceedings of the 35th AAAI Conference on Artificial Intelligence*, Virtual, pp. 4697-4705, 2021. https://doi.org/10.1609/aaai.v35i5.16600

Liqing Qiu an Associate Professor and Master's Supervisor, earned a Ph.D. in Computer Software and Theory from Beihang University in 2008. Specializing in data mining and social networks, Dr. Qiu has published over 20 papers as the first or corresponding author in renowned journals such as Information Science, Knowledge-based Systems, Journal of Network and Computer Applications, Soft Computing, Applied Intelligence, Journal of Information Science, and Data Analysis and Knowledge Discovery. Dr. Qiu has led research projects funded by the National Natural Science Foundation of China, Shandong Natural Science Foundation, Shandong Social Science Planning Project, National Postdoctoral Foundation, Shandong Postdoctoral Foundation, and Qingdao Social Science Planning Project. Additionally, Dr. Qiu has significantly contributed to more than 10 national and provincial key projects.

Si'ao Sun born in 1999, is currently an MS candidate at the School of Computer Science and Engineering, Shandong University of Science and Technology. Her current research interests include Neural Networks, Intelligent Education, and Deep Learning. She currently has one paper under external review and another under submission.