

MPHINE-KT: Heterogeneous Information Network Embedding with Meta-path Knowledge Tracing

Meizi Li, Zhijie Shao, Li Xu, Maozhen Li, Yunwen Chen, Ru Yang*, Bo Zhang*

Abstract—Contributions: This paper proposes a novel MPHINE-KT model for knowledge state modeling and academic performance prediction. It designs learner learning experience to mine sequential knowledge transfer, analyzes the individuality and commonality of learning experience and abilities via heterogeneous information networks, and introduces personal attributes to alleviate data sparsity, so as to enhance the prediction capability of knowledge tracing.

Background: Knowledge tracing models are widely adopted on online education platforms to analyze learning behaviors and predict learning outcomes for instructional feedback. However, current methods cannot effectively capture transfer relationships among knowledge points, ignore individual differences in learner attributes, and fail to integrate multi-attribute features, which leads to input data sparsity and limits the prediction accuracy of learners' future exercise performance.

Application Design: The model builds a behavior graph-based learning experience recognition module, constructs a learner heterogeneous information network, and feeds the extracted learner feature vectors into a GRU temporal prediction model for performance forecasting.

Intended Outcomes: It explores latent correlations among learners, learning experience and abilities, realizes accurate prediction of future exercise performance, and provides reliable support for personalized online education.

Findings: Experimental results on multiple real-world datasets demonstrate that the proposed MPHINE-KT model achieves superior performance against mainstream existing knowledge tracing models.

Index Terms—heterogeneous graph neural networks; knowledge tracing; graph attention; graph representation learning

I. INTRODUCTION

THE popularization of information technology has given rise to a new wave of online education, such as Intelligent Tutoring Systems (ITS) [1], [2] and Massive Open Online Courses (MOOCs) [3]–[5], which break the geographical and time constraints of traditional education. By tracking learners' learning, online education provides precise teaching

support, and stimulates learners' motivation. In the big data research of smart education, it is particularly important to establish a knowledge tracing (KT) [6] model. KT model aims to model the knowledge state of learners based on their historical interactions and predicts learners' learning outcomes accordingly [7]. The model analyzes the input data of students' historical answer interaction records sequence, which includes the question number and whether the question was answered correctly or not, to predict whether students will be able to answer a new question correctly or not when they encounter it in the future. This improves teaching models and learner assessment, and thus supports multiple key educational processes or scenarios such as automated evaluation [8], learning resource recommendation [9], educational interaction robots [10], and test paper generation [11].

Among the traditional KT models, the most representative one is the Bayesian Knowledge Tracing (BKT) [12]. The BKT model assumes that there are only two states of knowledge mastery for learners: mastery and non-mastery, and there is no correlation between each question. The limitations of the assumptions result in a BKT model that does not capture the underlying semantic information between exercises well, thus affecting the performance and accuracy of the model.

In recent years, deep learning has achieved good results in many fields, and many researchers have applied it to the field of KT. Among them, the most pioneering work is the Deep Knowledge Tracing (DKT) [13] model, which mainly adopts Recurrent Neural Network (RNN) [14] to simulate the learner's implicit knowledge mastery state.

After gradual improvement, the KT model has now embedded a variety of key factors affecting the prediction results, and thus becomes more in line with the actual educational scenarios. Liu et al. [15] proposed a KT architecture based on RNNs by using machine learning methods to capture learners' behavioral and practice characteristics. Sun et al. [16] extracted learner behaviors and question features from knowledge graphs and relational databases to obtain a more accurate knowledge background structure for subsequent tasks. These existing studies usually focus on the problem of how to deal with learners' learning record interaction data, but few of them comprehensively consider the problem of the impact of learners' learning experience, individuality, and commonality of learning abilities on the prediction results of KT. For example, there are learners who have rich learning experiences in algebra-based questions but have below-average learning abilities. Therefore, the joint influence of the two factors should be considered in the process of predicting learners' academic performance.

Meanwhile, to address the data sparsity problem in the DKT model, many scholars have tried to increase the input data on the basis of the original model to improve the prediction

This work was supported in part by the National Natural Science Foundation of China under Grant 62477032, Grant 62302306, Grant 62372300, and Grant 62201350, and in part by the Research Base of Online Education for Shanghai Middle and Primary Schools.

Meizi Li, Zhijie Shao, Li Xu and Ru Yang are with the college of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 200234, China (email: limeizi@shnu.edu.cn, szhijie2000@163.com, yangru@shnu.edu.cn).

Maozhen Li is with the College of Information, Mechanical, and Electrical Engineering, Shanghai Normal University, Shanghai 200234, China, and also with the Department of Electronic and Computer Engineering, Brunel University, UB8 3PH London, U.K. (e-mail: maozhen.li@brunel.ac.uk).

Yunwen Chen is with the DataGrand Inc., Shanghai 201203, China (e-mail: chenyunwen@datagrand.com).

Bo Zhang is with the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 200234, China, and also with Shanghai University of Electric Power (e-mail: zhangbo@shnu.edu.cn).

(Corresponding author: Ru Yang, Bo Zhang.)

effect of the knowledge tracing model. Zhang et al. [17] used a heterogeneous information network to model the answering questions and their attributes to solve the data sparsity problem of the learners in the actual learning scenarios. Lu et al. [18] considered the linkage of the concepts between courses and modeled the antecedent-successor relationship between concepts to solve the data sparsity problem in knowledge tracing.

However, these approaches fail to fully utilize the individual attribute features embedded in learners' question-answering behavioral patterns. A critical gap lies in the absence of integrating individual attribute features with behavioral features from question-answering activities, which is essential for capturing learners' unique differences and mitigating input data sparsity. Additionally, these methods lack the capability to distinguish between the diverse characteristics of learners' individual attributes and fail to comprehensively incorporate multi-attribute features, thereby exacerbating data sparsity issues. Consequently, this limitation hinders knowledge tracing models from accurately predicting learners' performance in future exercise answering.

Therefore, in order to solve the above shortcomings, this paper proposes a knowledge tracing method called Heterogeneous Information Network Embedding with Meta-path Knowledge Tracing (MPHINE-KT). First, GNNs [19] are used to obtain the embedded representations of the constructed behavioral graphs and perform clustering division to predict the distribution of learners' learning experience. Then, the learner heterogeneous information network [20] is constructed by combining the learners' learning experience and learners' learning ability. The embedding learning representation of heterogeneous information network is used to obtain the learner fusion feature embedding vector. Finally, the embedded vectors are used as additional inputs to the DKT model to solve the data sparsity problem in the DKT model and to improve the prediction effect of the knowledge tracing model in predicting learners' future performance.

The main contributions of this paper are:

- (1) It proposes a new KT model called MPHINE-KT to predict the accuracy of learners' future academic performance. The effects of individuality and commonality between learners' learning experience and learning abilities through heterogeneous information networks are analyzed, learners' individual attributes are introduced, and the problem of data sparsity is alleviated.
- (2) It introduces a learners' learning experience extraction module, which mines and analyzes the correlation between learners' behaviors and learners' learning experience, and thus provide support for the precise learners' learning experience service.

II. RELATED WORK

This paper focuses on exploring a knowledge tracing model based on heterogeneous graph neural network. Here this paper gives a literature review on heterogeneous graph neural network and knowledge tracing model, respectively.

A. Heterogeneous Graph Neural Network

Heterogeneous Graph Neural Networks (HGNNs) [21] aim to handle complex graph data containing multiple types of nodes and edges. Traditional graph embedding methods, such

as models based on matrix factorization and random walks, have issues like high time-space costs and difficulty in capturing high-order non-linear features, and most of them are designed for homogeneous graphs. In contrast, HGNNs can deal with heterogeneous graphs with various types of nodes and edges. Through specially designed neural network architectures, they can effectively capture the complex structures and semantic information in heterogeneous graphs. According to the differences in their learning architectures and technical applications, they can be mainly divided into three categories: autoencoder-based, convolution-based, and adversarial-based models.

Autoencoder-based HGNN models use encoders to learn node structures and attributes for generating embeddings and decoders to preserve the information of the original graph. For example, HIN2Vec [22] proposed by Fu et al. uses meta-paths and feed-forward neural networks to learn representations and optimize the model. SHNE [23] proposed by Zhang et al. is for heterogeneous graphs with text attributes. It uses meta-paths and GRUs to aggregate structural and semantic information.

Convolution-based HGNN models take the message-passing mechanism as the core. They aggregate neighbor information to update the embedding of the central node. For instance, HAN [24] proposed by Wang et al. designed node-level and semantic-level attention mechanisms to capture semantics. Inspired by HAN, Fu et al. proposed MAGNN [25]. When aggregating at the node-level, MAGNN aggregates all nodes on the meta-path to maintain the heterogeneity of the graph.

Adversarial-based HGNN models leverage the adversarial training of GANs to obtain more robust node embeddings. HeGAN [26] proposed by Hu et al. extended the discriminator and generator to make them relation-aware. MV-ACM [27] proposed by Zhao et al. is for incomplete heterogeneous graphs. Based on a multi-view architecture, it applies adversarial learning to capture the complementary information of nodes.

B. Knowledge tracing model

Traditional knowledge tracing models use machine learning approaches to assess the learner's learning state, and the most classic model is BKT. BKT constructs a Hidden Markov Model [28] for each question under each concept. The model not only ignores the existence of intermediate states in the hidden state, but also ignores possible correlations between concepts. To improve the performance of the BKT model, Bayesian Knowledge Tracing-Same Template (BKT-ST) [29] introduces Question Tagging and Individualized Prior Knowledge to obtain a representation of a question taking into account the correlation between questions. Multistate BKT (MS-BKT) [30] proposed a multi-state Bayesian Knowledge Tracing model that extends the learner's knowledge acquisition states to 21. This fine-grained classification of multiple states enables a more accurate assessment of the learner's learning state.

In recent years, the rise of deep learning has also led to advances in the field of knowledge tracing. Deep Knowledge Tracing (DKT) is the first model that proposes an attempt to employ a recurrent neural network (RNN) approach to knowledge tracing, which utilizes the LSTM [31] to model the temporal sequences of learner-question interactions. Dynamic Key-Value Memory Networks (DKVMN) [32] employs the key

and value external memory matrices [33] to come up with a way to capture long term dependencies of historical records. Deep-IRT [34] combines Item Response Theory (IRT) [35] with a deep neural network structure based on DKVMN for knowledge tracing modeling for the first time. Self Attentive Knowledge Tracing (SAKT) [36] and Attentive Knowledge Tracing (AKT) [37], which introduce attentional mechanisms [38] into the knowledge tracing domain, are used to capture the correlation between the current question and the historical interaction sequence.

In recent research, a number of scholars have combined GNNs with KT. In this way, a large amount of non-Euclidean data in the field of education can be transformed into structured data, so as to better extract the information of questions and concepts at the level of the graph structure. Graph-based Knowledge Tracing (GKT) [39] was the first method to use the graph neural network approach in the field of KT. Graph-based Interaction model for Knowledge Tracing (GIKT) [40] used GCN [41] to aggregate the features of neighboring nodes to get the representation of the target node, which overcame the problem of sparsity of question data. Joint graph convolutional network based deep Knowledge Tracing (JKT) [42] builds the connection graph between questions under different concepts and obtains the deep semantic information between questions and concepts. The KSGKT [43] model proposed by Gan et al. automatically infers the knowledge structure and integrates it into the KT process, combining graph representation learning with an attention mechanism. These models have been verified by experiments to be superior to traditional KT models, promoting the development of this field.

In recent years, heterogeneous graph structures have been widely applied in knowledge tracing tasks to alleviate the data sparsity problem. TCL4KT [44] proposed by Sun et al. constructs a weighted heterogeneous graph encompassing questions, skills, and students, optimizing question embeddings through three-view contrastive learning; HCMKT [45] proposed by Liu et al. designs coarse-grained and fine-grained dual views, enhancing representation quality by integrating meta-path contexts with contrastive learning; Wu et al. [46] focus on the programming education scenario and construct a heterogeneous graph consisting of knowledge concepts, programming questions, and student codes. To address the performance issues in knowledge tracing caused by the heterogeneity of knowledge structures, HHSKT [47] model proposed by Ni et al. traces knowledge based on heterogeneous hierarchical differentiation. It constructs heterogeneous graph components, transforms the TrueSkill system, and utilizes window attention. Jiang et al. proposed a model [48] that adopts a meta-path-based embedding pre-training method, constructs a heterogeneous graph to obtain node embeddings, and integrates them into existing models. S^2 -HHN [49] model constructs a heterogeneous hypergraph network in combination with self-supervised learning, designs attention mechanisms, and introduces data augmentation and contrastive learning. However, these methods still have two shortcomings: first, most of them construct networks centered on "question-skill", ignoring the dynamic impact of learners' individual attributes (such as learning experience and learning ability) on knowledge states; second, they fail to fully explore the common associations among learner groups (such as interactions between similar learning experiences and comparable learning abilities).

To address the data sparsity problem in the DKT model,

many scholars have tried to add input data to the original model to improve the prediction effect of the knowledge tracing model on learners' future performance. Attention-based Knowledge Tracing with Heterogeneous Information Network Embedding (AKTHE) [17] employs heterogeneous information network modeling to answer questions and their attributes, which solves the problem of data sparsity of learners in real learning scenarios. Concept Map Driven Knowledge Tracing (CMKT) [18] considers the linkage of concepts among the courses and models the antecedent-successor relationship among the concepts as an ordered-pair structure, which solves the problem of data sparsity in knowledge tracing.

However, there are still problems: (1) They fall short in extracting the features of transfer relationships between knowledge points for learners. (2) They also struggle to differentiate individual learner attributes and fail to fully integrate learners' multi-attribute features, leading to sparse input data..

III. PROBLEM DEFINITION

In the knowledge-tracing task, supposing a set of learners $S = \{s_1, s_2, \dots, s_n\}$ and a set of questions $Q = \{q_1, q_2, \dots, q_m\}$ of online learning history answer records, given a learner s_i 's history of learning answer sequence $rec_i = \{r_1, r_2, \dots, r_{t-1}\} = \{(q_1, a_1, e_i), (q_2, a_2, e_i), \dots, (q_{t-1}, a_{t-1}, e_i)\}$, where $q_i \in Q, a_i \in \{0, 1\}$, t denotes the learner's interaction order, e_i denotes the learner s_i 's embedding vector of the learner heterogeneous information network.

Definition : The task of knowledge tracing is defined as to model the knowledge state of learners and predict the possibility of learners answering the exercise correctly at the next time step $t+1$ as $P(a_{t+1} = 1 | q_{t+1}, rec_i)$. In order to predict the learners' correctness in answering the questions at the next time stamp, the model need to extract a representation of the input questions. Since the number of questions in the existing learners' online answer records is relatively large compared to the number of concepts, directly using the one-hot encoding of the question numbers to represent the questions will lead to the problem of the resulting vectors being too high dimensional and sparse. In this paper, MPHINE-KT assumes that the questions containing the same number of concepts and containing the same concepts are all regarded as the same question, and then all nodes are one-hot encoded to the vector k_i which can be obtained as the set of concepts $K = \{k_1, k_2, \dots, k_M\}$. To better mine the learner's historical answer sequence data, questions is represented as q_i , which is obtained by making vector product of k_i and difficulty coefficients of the questions. The formula of the representation of the question i is as follows:

$$q_i = k_i \bullet difficulty_i \quad (1)$$

The difficulty coefficient of a question is the value reflecting the degree of difficulty of the question. It is expressed as the number of people who answered the question correctly as a percentage of the total number of people. The formula is shown as below:

$$difficulty_i = \frac{t_i}{N_i} \quad (2)$$

where t_i denotes the number of learners who answered question i correctly, and N_i denotes the total number of learners who answered question i . The difficulty coefficients of each

TABLE I
KEY NOTATIONS AND DEFINITIONS.

Notation	Definition
$S = \{s_1, s_2, \dots, s_n\}$	Set of learners
$Q = \{q_1, q_2, \dots, q_m\}$	Set of Exercises
$rec_i = \{r_1, r_2, \dots, r_t\}$	The historical answer sequence of a learner
q_t	The embedding vector of the exercise
$K = \{k_1, k_2, \dots, k_m\}$	Set of knowledge points
$G_b = (V_b, E_b)$	Learner behavior graph
e_{ij}	The edge connecting node q_i to node q_j
d_{ij}	The number of edges from node q_i as the starting node to the terminating node q_j reflects the strength of the correlation between node q_i and node q_j
$G_{l-HIN} = (V_G, E_G)$	Learner Heterogeneous Information Network
$V_b = \{e, a, s\}$	The set of nodes in the learner heterogeneous information network includes student (s) and the learners' learning experience (e) and learning ability (a)
$E_b = \{e_b^{SAS}, e_b^{SES}\}$	The set of edges in the learner heterogeneous information network includes student-learning experience-student(SES) meta-path and the student-learning ability-student''(SAS) meta-path
$\{\phi_1, \phi_2\}$	Node Types in the Learner Heterogeneous Information Network
$\{P_1, P_2\}$	Meta - path Types in the Learner Heterogeneous Information Network
β_{ϕ_k}	Contribution Values of Different Types of Neighbor Nodes to the Target Node
β_{p_n}	Contribution Values of Neighbor Nodes along Different Meta - paths to the Target Node
z_i^{AC}	Node - level Attention Embedding Vector
z_i^p	Meta - path Level Attention Embedding Vector

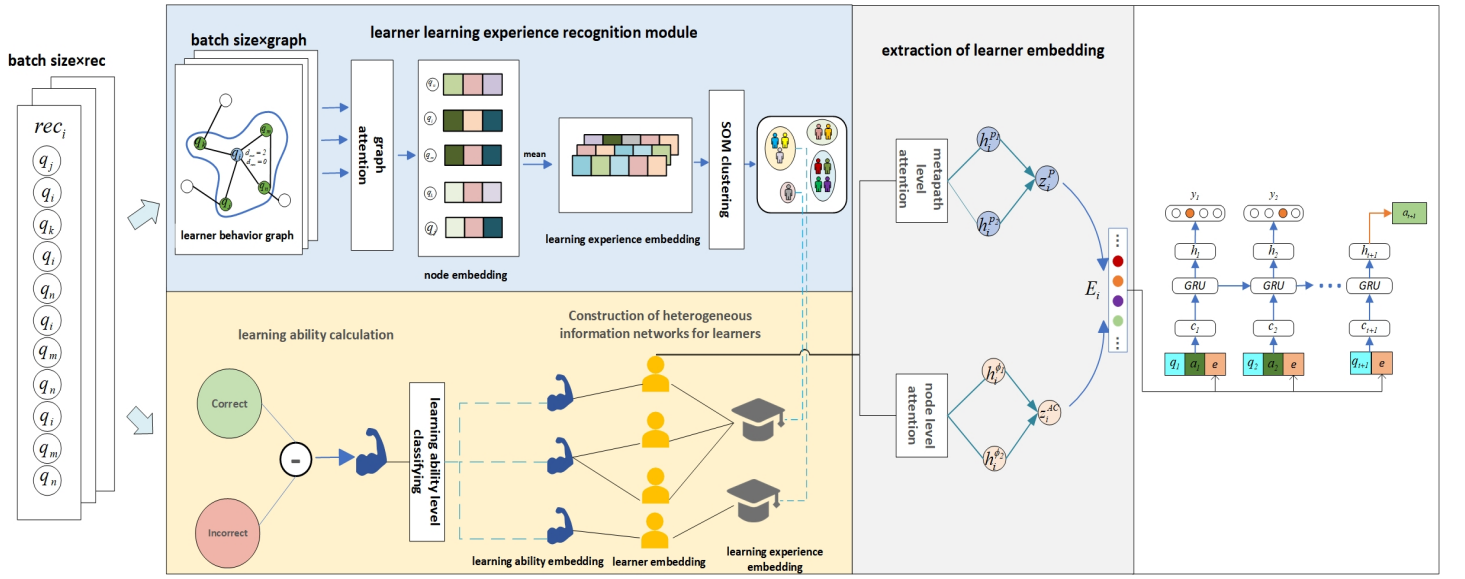


Fig. 1. The MPHINE-KT framework overview

question range from 0 to 1. The larger the difficulty coefficient value, the easier the question, and the smaller the difficulty coefficient, the more difficult the question.

IV. MODEL

A. Overview of the model

This paper introduces a novel knowledge tracing (KT) model named MPHINE-KT, as visualized in Fig. 1. The model comprises two sequential modules designed to capture learners' comprehensive profiles. The first module, Learner Learning Experience Recognition, processes learners' answer sequences to construct behavior graphs, where graph attention mechanisms extract learning experience vectors. These vectors are then clustered to categorize learners' learning patterns. The

second module, Heterogeneous Information Network Embedding, computes learning abilities from the answer sequences and fuses these ability attributes with the clustered learning experience categories (output from the first module) to build a heterogeneous information network. Node embeddings are then extracted from this network to form final learner representations. The final learner representation obtained through this fusion is then utilized in a time-series prediction network to forecast learners' future learning performance.

B. learning experience recognition based on Learner behavior graph

This section introduces the method of learning experience recognition based on learner behavioral graph, which firstly

performs the construction of learner behavioral graph, and then extracts of learner behavioral graph embedding, and finally completes the SOM clustering [52]. The extracted learners' learning experience will be used in the next step of learner heterogeneous information network construction.

1) *Construction of learner behavior graphs*: The graph construction module is to transform the sequence of learners' question answering behaviors into the structure of a graph for subsequent work.

Based on the set of m test questions $Q = \{q_1, q_2, \dots, q_m\}$ and learners' answer sequences $rec_i = \{r_1, r_2, \dots, r_{t-1}\}$, a learner behavior graph can be constructed as $G_b = \{V_b, E_b\}$, where each node $v \in V_b$ denotes a question answered by the learner during the question answering process $q \in Q$ and each edge $e_{ij} \in E_b$ represents the correlation between q_i and q_j . At the same time, MPHINE-KT uses d_{ij} to denote the relationship between a node q_i as the starting node to the terminating node q_j . The number of d_{ij} reflects the strength of the correlation between node q_i and node q_j . Given a learner behavior graph G_b , the edge weights in the learner behavior graph can be viewed as the direction of transfer diffusion of the learner's learning experience. The learners' learning experience propagation of other nodes with the target node is captured as the degree of other nodes with the target node. The edges flowing out from the concept nodes are the transfer diffusion of learners' learners' learning experience, while the edges flowing in from the concept nodes are the aggregation of learners' learning experience.

For example, suppose that the learner answer sequence is $q_{\{1,2,\dots,t\}} = q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_4 \rightarrow q_3 \rightarrow q_2 \rightarrow q_4 \rightarrow \dots$, with $d_{12} = 1$ denoting the number of edges of node q_1 to node q_2 , and $d_{34} = d_{43} = 1$ denoting that the number of edges from node q_3 to node q_4 is the same as the number of edges from node q_4 to node q_3 edges are both 1.

Specifically, first, the exercise embedding vector is obtained by multiplying the exercise difficulty coefficient by the knowledge point vector, which realizes the correspondence between exercises and knowledge points. Then MPHINE-KT considers the correlation of the nodes of the questions to construct the learner behavior graph. In the entitled undirected graph, the structure of the graph reflects the learner's choices in the answering process. Therefore, the questions in the sequence of learners' answering behaviors are obtained.

2) *Learner Behavior Graph Embedding Extraction*: As different exercises exert varying impacts on learners, ordinary GNNs typically treat all nodes within the neighborhood uniformly. In contrast, the graph attention mechanism has the capability to autonomously discern which neighborhood nodes are more crucial for updating the features of the central node. By adapting to dynamic changes in the relationships between nodes, it adaptively focuses on important information, thereby learning the representations of exercise nodes with different characteristics. Therefore, this paper employs the graph attention mechanism to effectively represent the structural data inherent in graphs. The inputs to the attention module are questions nodes and connections between nodes. Firstly, the graph attention network is used to calculate the attention values of node q_i in the learner's behavior graph and assign different weights to different neighbor nodes. The formula is shown

below:

$$\alpha_{ij} = \frac{\exp(\text{LeakyRelu}(a^T [Wq_i || Wq_j]))}{\sum_{q_k \in V_b} \exp(\text{LeakyRelu}(a^T [Wq_i || Wq_k]))} \quad (3)$$

where the weight matrix W and parameter α are trainable parameters, LeakyRelu is the activation function, $||$ represents vector concatenation, and $\exp()$ represents the exponential function.

Then, to combine the obtained information about the attributes of the nodes of the learner behavior graph and the information about the topology of the behavior graph, the attention between different nodes and the attention between edges of different nodes in the graph attention network can be combined to form the attention coefficient, which is calculated as follows:

$$\theta_{ij} = \frac{d_{ji}}{d_i} \quad (4)$$

$$g_{ij} = \alpha_{ij} \theta_{ij} \quad (5)$$

Where d_{ji} is the the number of edges from node q_j to node q_i , and d_i is the number of all edges that point to node q_i .

Based on the attention coefficients captured by the graph attention network, the feature vectors of the nodes in the behavioral graph are updated. Then employing multi-head attention can enhance the updating and representation learning of node features. Finally averaging is employed for the aggregation of relevant information to obtain the behavioral graph based embedding I . The formula is shown below:

$$q_i = \sigma \left(\sum_{q_i \in V_b} \sum_{q_j \in N_i} W_i g_{ij}^h q_j \right) \quad (6)$$

$$q_i(H) = ||_{h=1}^H \sigma \left(\sum_{q_i \in V_b} \sum_{q_j \in N_i} W_i^h g_{ij}^h q_j \right) \quad (7)$$

$$I = \text{mean} \left(\sum_{q_i \in V_b} q_i(H) \right) \quad (8)$$

Where σ is the sigmoid function, W_i is the trainable parameter, H is the number of heads of attention, and W_i^h denotes the weight parameter of the h th attention head, $\text{mean}()$ represents taking the average value.

3) *SOM clustering*: After acquiring the representations I of learning experience for all learners, MPHINE-KT must cluster these representations to facilitate the construction of a heterogeneous information network for learners and to extract learner representations effectively. Learners with similar learning experience are clustered together to obtain the learning experience category for each learner. Since SOM clustering can handle various complex data distributions and maintain the topological structure of the data, MPHINE-KT chooses the SOM algorithm [52] for clustering. First, randomly initialize the neuron weights W in the SOM, and perform vector normalization on the input vectors I and the weights W . Then, enter the competition process. Calculate the Euclidean distance between the normalized vector $\hat{I}$ and each neuron, and select the neuron with the shortest distance as the winning neuron. At the same time, update the weights of the winning neuron and the neurons in its neighborhood. The formula is

shown below:

$$\hat{I} = \frac{I}{\|I\|} \quad (9)$$

$$\hat{W} = \frac{W}{\|W\|} \quad (10)$$

$$\hat{W}_j(m+1) = \hat{W}_j(m) + (m, dis)(I - \hat{W}_j(m)) \quad (11)$$

Among them, $\| \cdot \|$ represents the Euclidean norm, j represents the serial number of the neuron that needs to be updated, and η is the learning rate related to the number of training iterations m and the topological distance dis of the winning neural unit. If the preset number of iterations is reached, the network training process is terminated, and finally the neuron closest to the input data is the result of clustering. After obtaining the learners' learning experience category of each learner, this information is incorporated as a discrete custom constant to the learner's features, which is used for the construction of the learner heterogeneous information network in next module.

C. Metapath-based knowledge tracing of learners' heterogeneous information networks

In the previous section, MPHINE-KT constructs the learner behavior graph, extracted graph embeddings to obtain learners' learning experience. These experience were then clustered to get the learning experience category for each learner. To explore the relationships between learners and their learning experience, as well as other learner-related attributes, MPHINE-KT designed the metapath-based knowledge tracing of learners' heterogeneous information networks. This section introduces the metapath-based knowledge tracing of learners' heterogeneous information networks. This approach first involves constructing the learner heterogeneous information network. Subsequently, it extracts the metapath-based embeddings of the learner heterogeneous information network, which takes into account the influences of learners' individual attribute characteristics and learning abilities. Finally, these embeddings are added as input to the time-series prediction module, thus improving the accuracy of predicting learners' future learning performance

1) *Learner Heterogeneous Information Network Construction*: The question-answering behavior sequence not only implies the learners' learning experience, but also reflects learners' learning ability over time. The learners' learning ability can be used to differentiate the individual learning ability differences of different learners.

Learning ability is expressed by calculating the difference between the proportion of correct answers $Correct(q_{ij})$ and the proportion of incorrect answers $Incorrect(q_{ij})$, and is characterized by $Ability$, which is calculated by the following formula:

$$Correct(q_{ij}) = \frac{\sum_j^{|N_{ij}|} \{b_{ij} == 1\}}{|N_{ij}|} \quad (12)$$

$$Incorrect(q_{ij}) = \frac{\sum_j^{|N_{ij}|} \{b_{ij} == 0\}}{|N_{ij}|} \quad (13)$$

$$Ability(s_i) = \begin{cases} Correct(q_{ij}) - Incorrect(q_{ij}), & |N_{ij}| \geq 5 \\ 0.5, & \text{else} \end{cases} \quad (14)$$

Where N_{ij} represents the number of learner s_i attempts to answer the question q_j . $Correct(q_{ij})$ and $Incorrect(q_{ij})$ represent the number of times the learner s_i correctly or incorrectly answered the question q_j . b_{ij} represents the probability that the learner answered the question correctly in moment t , which takes the value of 0 or 1, with 0 representing an incorrect answer and 1 representing a correct answer. Ability takes values between -1 and 1.

In this paper, MPHINE-KT divides the learners into 5 ability level groups, calculate the average learning ability of each ability level group, and then assign the learners to the average learning ability value of this ability level group, as shown in Table II.

TABLE II
LEARNER LEARNING ABILITY LEVEL CORRELATION

Learning ability level	Average learning capacity
Poor	≤ 0.5
Fair	0.5-0.7
Average	0.7-0.8
Good	0.8-0.9
Excellent	0.9-1.0

The learner heterogeneous information network is then constructed to represent the relationship between the student (s) and the learning experience (e) and learning ability (a). Two types of connections are established for the three objects: the "learner-learning experience" connection and the "learner-learning ability" connection. The objects in the learner heterogeneous information network can be connected through the defined connections, containing the "student-learning experience-student" (SES) meta-path and the "student-learning ability-student" (SAS) meta-path. Then, the learner heterogeneous information network is formulated as $G_{L-HIN} = (V_G, E_G)$, where $V_G = \{s, e, a\}$ denotes the set of student, learning experience, and learning ability nodes, and $E_G = \{e_b^{SES}, e_b^{SAS}\}$ denotes the set of edges corresponding to the SES and SAS meta-paths. The schematic diagram of the learner heterogeneous information network is shown in Fig. 2.

2) *Metapath-based embedding of heterogeneous information networks for learners*: Heterogeneous information network embeddings with multiple semantics are extracted by using a hierarchical attention mechanism for heterogeneous information network embeddings of learners at the node level

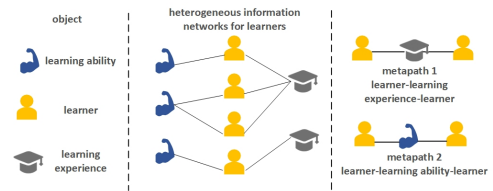


Fig. 2. The construction of a learner heterogeneous information network includes three types of heterogeneous nodes: learning ability, learner, and learning experience. Two types of metapaths: student-learning experience-student (SES), student-learning ability-student (SAS)

and meta-path level attention mechanism, respectively. The node feature vector's dimensionality transformation is required and thus the projected features of all nodes share the same dimensions, as shown in the following equation:

$$h_i = W_{O_P} \bullet x_i^A \quad (15)$$

Where $h_i \in R^{d_{low}-d}$ is the projected implicit feature vector, and W_{O_P} is the learnable weight matrix, and $x_i^A \in R^{d_A}$ are the original feature vectors of different types of nodes.

Node weights are updated using node level attention mechanism. For the same type of neighboring nodes with different contributions to the target node, the node's attention score is calculated as shown below:

$$h_i^{\varphi_k} = \sigma \sum_{j \in N_i^{\varphi_k}} \alpha_{ij} \bullet h_j \quad (16)$$

$$\alpha_{ij} = \frac{\exp(\text{LeakyRelu}(a_{type}^T [Wh_i || Wh_j]))}{\sum_{h_m \in N_i^{\varphi_k}} \exp(\text{LeakyRelu}(a_{type}^T [Wh_i || Wh_m]))} \quad (17)$$

where a_{type} is a vector of same type node level attention parameters, is a learnable parameter, and α_{ij} is the importance weight of nodes of the same type to the target node, and $h_i^{\varphi_k}$ is the updated vector of nodes of the same type. Considering that the contribution of different types of nodes to the target node is also different, after completing the calculation of the weight of the attention of the same type of nodes to the target node, the weight of the attention of each node type to the target node is calculated as shown in the formula below:

$$w_{\varphi_k} = \frac{1}{|V|} \sum_{i \in V} \alpha_{node_q} \bullet \tanh(W_{node_q} h_i^{\varphi_k} + b_{node_q}). \quad (18)$$

$$\beta_{\varphi_k} = \frac{\exp(w_{\varphi_k})}{\sum_{k=1}^2 \exp(w_{\varphi_k})} \quad (19)$$

$$z_i^{AC} = \sum_{k=1}^2 \beta_{\varphi_k} \bullet h_i^{\varphi_k} \quad (20)$$

where α_{node_q} are different types of node-level attention parameter vectors. W_{node_q} and b_{node_q} are the weight matrix and offset values, respectively, both of them are learnable parameters. w_{φ_k} is the updated vector obtained from the above equation the linear variation values of $h_i^{\varphi_k}$ for calculating the weights of each type of node. The obtained w_{φ_k} is normalized to get β_{φ_k} . β_{φ_k} denotes the contribution value of different types of neighboring nodes to the target node. Then $h_i^{\varphi_k}$ and β_{φ_k} are multiplied to obtain the embedded node level vector of the target node z_i^{AC} .

In the learner heterogeneous information network, the SES meta-path is employed to represent scenarios in which different learners exhibit similar learning experience. Conversely, the SAS meta-path simulates situations where learners possess comparable learning abilities. These two distinct meta-paths contribute differently to the target node due to their varying implications. Consequently, a meta-path-level attention mechanism is implemented to learn the importance of different meta-path instances when influencing the target node. Two symmetric metapaths are defined in the learner heterogeneous information network $E_b = \{e_b^{SAS}, e_b^{SES}\}$. Considering different metapaths contribute differently to the target node, the

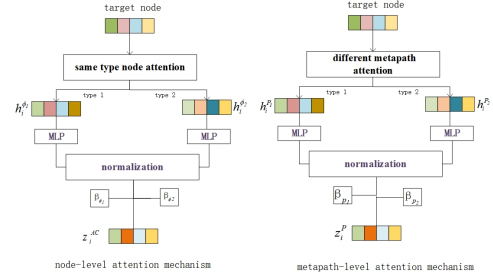


Fig. 3. schematic diagram of node-level and metapath-level attention mechanism

metapath-level attention mechanism is used for path-to-target node messaging. First, node update is performed for the nodes in the metapaths with the following formula:

$$h_i^{p_n} = \frac{1}{d_i + 1} h_i + \sum_{j \in N_i^{p_n}} \frac{1}{\sqrt{(d_i + 1)(d_j + 1)}} h_j \quad (21)$$

Where $h_i^{p_n}$ is the updated node vector on the meta-path, and d_i and d_j are the degrees of node i and node j , and $N_i^{p_n}$ is the neighbor node of target node i in different meta-path cases.

Then, MPHINE-KT calculates the target node i 's attention score with all the meta-paths $E_b = \{e_b^{SAS}, e_b^{SES}\}$ and then obtain the feature vector of the target node based on the meta-path level by normalization operation. The formula is shown as below:

$$w_{p_n} = \frac{1}{|V|} \sum_{i \in V} \alpha_{path_p} \bullet \tanh(W_{path_p} h_i^{p_n} + b_{path_p}) \quad (22)$$

$$\beta_{p_n} = \frac{\exp(w_{p_n})}{\sum_{n=1}^2 \exp(w_{p_n})} \quad (23)$$

where α_{path_p} is a vector of meta-path level attention parameters, W_{path_p} and b_{path_p} are the weight matrix and offset values, respectively. w_{p_n} are the linear variation values obtained according to $h_i^{p_n}$. Then MPHINE-KT normalizes $h_i^{p_n}$ to obtain β_{p_n} , and β_{p_n} is used to represent the contribution value of neighboring nodes on different meta-paths to the target node. Then the values of β_{p_n} and $h_i^{p_n}$ are multiplied to obtain the embedded node level vector of the target node z_i^P .

The obtained node level attention embedding vectors and meta-path level attention embedding vectors are spliced as the final embedding representation of the learner's heterogeneous information network. The formula is shown below:

$$E = z_i^{AC} \oplus z_i^P \quad (24)$$

Among them, $\oplus$ represents the concatenation operation, and i represents the serial number of the learner.

3) *Time series forecast*: The proposed model takes the learner's historical question-answering behavior data and learner's embedding vector as the input. First, the raw data answering interaction records (q_t, a_t) and the embedding vector e of the learner's heterogeneous information network are spliced to obtain the spliced ternary c_t . The formula is shown below:

$$c_t = \text{Concat}(q_t, a_t, e) \quad (25)$$

Since the sequence data of students' answering behaviors has the characteristics of time series, the Gated Recurrent Unit (GRU) is a neural network structure specifically designed for processing sequential data. It can effectively capture long-term dependencies in the sequence. The GRU can automatically extract valuable features from the input sequence data of learning behaviors and learn the hidden patterns in the data. The data is passed through the GRU neural network computation to the output layer and the output y_t is a numerical value that predicts the probability of the learner answering the new question correctly. The related formula is shown below:

$$y_t = \sigma(W_{out}^t h_t + b_{out}^t) \quad (26)$$

Where W_{out}^t is the output weight matrix and the bias term b_{out}^t is trained parameters of the model.

The training model is to predict the probability that the learner will answer the question correctly, and thus backpropagation can be performed based on a loss function consisting of the predicted score and the actual score (right answer 1/ wrong answer 0). The cross-entropy function is calculated, and the loss function for the entire sequence of questions answered by a single learner based on the predicted possibility of correctness y_{t+1} and the true label a_{t+1} is shown in the following equation:

$$Loss = -\sum_{t=1}^l (a_t \log \hat{y}_t + (1 - a_t) \log (1 - \hat{y}_t)) \quad (27)$$

According to the loss function, the model calculates the gradients of the learnable parameters of all modules based on the loss function values and update all learnable parameters until all learnable parameters converge iteratively on the dataset to obtain the knowledge tracing model.

V. RESULT

A. Dataset

The experimental data required for the MPFINE-KT proposed in this paper relies on four publicly available educational datasets. These four experimental educational datasets are ASSISTments2009 (Assist09), ASSISTments2015 (Assist15), [51] Algebra2005 (Algebra05) and Bridge2006 (Bridge06), which include learner stage-by-stage question-answer record data. These datasets are intended to reflect the real behavior and performance of learners in the learning process.

Assist09: (Available at: <https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data>. Accessed: May 14, 2026.) The Assist09 dataset is from the ASSISTments, which is an e-learning system for the 2009-2010 school year. There were duplicate records in the original Assist09 dataset, and after deleting and de-duplicating the duplicate records, a total of 4,217 learners answered the questions on 123 concepts in the remaining data.

Assist15: (Available at: <https://sites.google.com/site/assistmentsdata/home/2015-assistments-concept-builder-data>. Accessed: May 14, 2026.) The Assist15 dataset is collected on the same online education platform as the Assist09 dataset, recording data from the 2015-2016 school year period, with each question containing only one relevant concept. The Assist15 dataset contains a total of 19,917 learners answering questions on 100 concepts.

Algebra05: (Available at: <https://pslcdatashop.web.cmu.edu/KDDCup>. Accessed: May 14, 2026.) The Algebra05 dataset is

TABLE III
THE STATISTICS OF DIFFERENT KT DATASETS

dataset	Number of learners	Number of concepts	Total number of records
Assist09	4217	123	346860
Assist15	19917	100	708631
Algebra05	574	112	809694
Bridge06	1146	493	3679199

a sub-dataset of the KDD Cup2010 dataset and contains a total of 574 learners' answer records on 112 concepts, including detailed step-level learner responses.

Bridge2006: (Available at: <https://pslcdatashop.web.cmu.edu/KDDCup>. Accessed: May 14, 2026.) The Bridge2006 dataset is a sub-dataset of the KDD Cup2010 dataset, which contains the records of 1146 learners answering questions on 493 concepts, and the questions in the dataset are clearly labeled with the corresponding concepts.

Table III summarizes some basic information about the dataset, including the name of the dataset, the number of learners, the number of concepts, and the total number of response records. In addition, in this paper, these datasets are processed and divided for training and testing of knowledge tracing models. Statistics are performed for the training and test sets of question-answering behavior records in the datasets, and the datasets are partitioned based on the random order when training the models. The statistical results are shown in Fig. 4.

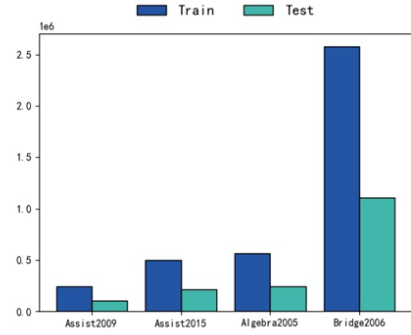


Fig. 4. Divisions of training and test sets for each knowledge-tracking dataset

B. Baseline Methods and parameterization

In this paper, comparative experiments were conducted to demonstrate the effectiveness of the proposed MPFINE-KT. The model proposed in this paper is compared with the following representative methods:

DKT [13]: The first model applies deep learning methods to the domain of knowledge tracing, using a recurrent neural network model to model the learner's knowledge state. The dropout of the model is 0.2, the hidden dimension of the neural network LSTM layer is set to 200, and the learning rate is set to 0.001.

DKT+ [52]: The model was proposed to augment the original DKT loss function with two additional regularization terms to address the limitations in DKT's ability to reconstruct an answer input and reduce inconsistency of answer prediction for questions sharing similar Knowledge points. The hidden dimension is set to 200, and the learning rate is set to 0.01, the

hyperparameter λ_r, λ_{w_1} and λ_{w_2} are set to 0.10, 0.03, 1.0, respectively.

GKT [39]: Considering that previous knowledge tracing models based on deep learning methods ignore the existence of underlying knowledge structures, GKT transforms knowledge structures into graphs, considers the correspondence between concepts and questions, and reconstructs the knowledge tracing task as an application of graph neural networks. The hidden dimension of the model is 64, dropout is 0.5, and the learning rate is set to 0.01.

AKT [37]: Due to the presence of diverse attributes in the learner answer data, and considering the missing learning features, AKT extracts guess and miss behavioral features from the question text, and uses the features for hidden state updating. The dropout of the model is 0.05, the dimension of the fully connected layer is set to 512, and the learning rate is set to 0.001.

SAKT [36]: The problem of sparse data in previous models is solved by migrating the application of the transformer encoder structure. The dropout of the model is 0.2, the multiple attention is set to 8, and the learning rate is set to 0.001.

DKVMN [32]: This model is adapted from the memory enhancement network to increase the interpretability of the model by calculating the attention between questions and concepts. The dropout of the model is 0.05, the embedding layer dimension is set to 200, and the learning rate is set to 0.001.

DEEP-IRT [34]: DEEP-IRT is a model based on IRT theory, combined with item response theory, and an explanatory model constructed on the basis of DKVMN. The dropout of the model is 0.05, the embedding layer dimension is set to 200, and the learning rate is set to 0.001.

GIKT [40]: GIKT constructs a problem-skill graph and leverages GCN to aggregate neighbor information, thereby enriching problem representations. The embedding dimensions of skills, problems and answers are set to 100. The hidden layer dimensions of the stacked two-layer LSTM are 200 and 100 respectively. The maximum aggregation layer of graph information is 3, the dropout rate is 0.8, the learning rate is set to 0.001, and the batch size is 32.

TCL4KT [44]: TCL4KT builds a weighted heterogeneous graph that incorporates students, problems and skills, and adopts a three-view contrastive learning framework to learn high-quality problem embeddings. The problem embedding dimension is 128, the temperature parameter τ in the contrastive loss is 0.8, the dropout rate is 0.3, the learning rate is set to 0.008, and the batch size is 80.

DGEKT [53]: DGEKT establishes a dual-graph structure consisting of a hypergraph and a directed graph, and introduces online knowledge distillation to fuse the two graph-based models. The output dimension of the two-layer GNN is 256, the hidden layer dimension of the GRU is 256, the temperature parameter μ in knowledge distillation is 0.5, the dropout rate is 0.2, the optimizer learning rate is 0.001, the weight decay is $1e-4$, and the batch size is 64.

For the model proposed in this paper, in order to guarantee the comparability of the experiments, the control variable method is used: the values of the learning rate parameter are set to $lr = 0.01$, $lr = 0.001$, $lr = 0.0001$, and $lr = 0.00001$, respectively. The same network structure model was run on the same hardware, and carry out the experiments of comparing the learning rate on different datasets. Fig. 5 gives the results

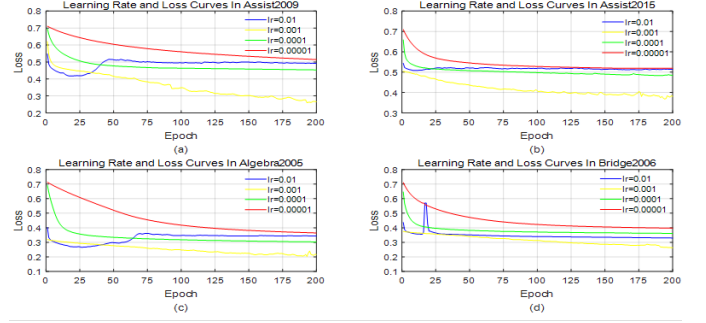


Fig. 5. Loss Function Optimization for Different Learning Rates on Various Knowledge Tracing Datasets

of the experiments. The results of the experiment show that as the model runs, the convergence of the function becomes slower; when $lr = 0.01$, the curve produces violent fluctuations and the loss value is relatively high; when $lr = 0.0001$ and $lr = 0.00001$, it can be seen that the curve is smoother, and the curve with $lr = 0.0001$ enters the convergence state earlier; when 0.001, the loss is in a range of small fluctuations, the loss value is relatively low. Thus, $lr=0.001$ is chosen as the optimal learning rate of the model. At the same time, for all datasets, all the learning records of learners are sorted according to the timestamp. In this paper, the training set and the test set are divided in a ratio of 8:2 from the dimension of time series. In this paper, the time series of the first 80% of all learners are used as the training set, and the remaining series are used for testing. During the model training, this paper uses the answering behavior sequence data in the training set to extract the learning ability of learners and the difficulty of exercises, and tests its generalization ability on the test set. Then, all input sequences are set to a fixed length of 100. For sequences longer than this fixed length, they are divided into several segments based on the fixed length. For those shorter sequences, they are padded to the fixed length with zero vectors. The uniform distribution is used to randomly initialize all parameters, and uses the Adam optimizer. The dimension of the embedding layer are set to 32, the batch size to 256, the hidden layer dimension to 64, the dropout rate to 0.5. To enhance implementation efficiency and boost parallelization performance, when aggregating features of the learner behavior graph via GAT, the graph data sampling approach in GIKT was adopted as a reference. Specifically, a fixed number of question neighbors is sampled for each batch of the behavior graph. During the process of predicting the learners' learning performance, each sample is run multiple times, and the model outputs are averaged to obtain stable prediction results. In this paper, the behavioral graph of each learner in each batch is first constructed, the representation of the learner's learning experience is extracted, and clustering is carried out. Then, the clustering results are combined with the calculated learning ability attributes to construct a learner heterogeneous information network for each learner in each batch. The representation of this heterogeneous network is extracted, so as to obtain the learner representation that integrates various learner characteristics. The number of neurons in the SOM clustering is initialized and tuned according to different datasets to achieve the best model performance. For the knowledge point-level baseline models, the exercise-knowledge point

TABLE IV
COMPARISON OF PREDICTION PERFORMANCE

Model	AUC				ACC			
	Assist09	Assist15	Algebra05	Bridge06	Assist09	Assist15	Algebra05	Bridge06
DKT	0.7661	0.727	0.8821	0.751	0.7354	0.7345	0.8556	0.752
DKT+	0.8327	0.7310	0.9151	0.8026	0.7782	0.7564	0.8616	0.849
GKT	0.8268	0.719	0.9152	0.7784	0.7783	0.7555	0.8589	0.8423
AKT	0.8239	0.7261	0.9012	0.8028	0.7769	0.7510	0.8531	0.8455
SAKT	0.7587	0.7037	0.9062	0.7636	0.7321	0.7510	0.8532	0.8413
DKVMN	0.8234	0.7279	0.9146	0.7996	0.7732	0.7561	0.8593	0.8487
DEEP-IRT	0.8224	0.7254	0.9116	0.7958	0.7729	0.7563	0.8584	0.8476
GIKT	0.8262	0.7267	0.9148	0.8123	0.7835	0.7542	0.8605	0.8489
TCL4KT	0.8375	0.733	0.9105	0.8136	0.7842	0.7457	0.8624	0.849
DGEKT	0.8356	0.7351	0.9133	0.8085	0.7856	0.754	0.8597	0.8475
MPHINE-KT	0.8501	0.7364	0.9229	0.8167	0.7876	0.7584	0.8637	0.8505

correlation table provided with the dataset is utilized, and regard the knowledge point mastery level as the probability of answering an exercise correctly. All the experimental results use server resources to train the model, and the server used is PowerEdge-T640, with 4 NVIDIA A40 GPUs.

C. Comparative Experimental Results

The results of the comparison experiments are shown in Table IV. It is noted that the AUC and ACC metrics of the MPHINE-KT are higher than those of the other benchmark models on the four datasets, and the performance on the Assist09 and Algebra05 datasets is particularly outstanding. Specifically, the AUC metrics of the MPHINE-KT are 1.26% and 0.88% higher than those of the optimal benchmark model on the Assist09 and Algebra05 datasets, while the ACC metrics are 1.19% and 0.18% higher than those of the optimal benchmark model on the Assist15 and Bridge06 datasets, respectively. Overall, the MPHINE-KT outperforms the other benchmark models on these four datasets.

Based on the experiments, the following conclusions are drawn. First, compared with the benchmark model, the MPHINE-KT is ahead of the other methods in terms of AUC and ACC on all four datasets, with higher accuracy of AUC and ACC, indicating that the it has better prediction performance. Second, compared with the DKT model, the MPHINE-KT has different degrees of performance improvement on the four datasets. The result indicates that the introduction of learner heterogeneous information network in the MPHINE-KT can improve the prediction accuracy of the model. Third, it can be found that the experimental results of each model in the dataset Assist15 are generally lower than the performance of other datasets, which may be related to the sparsity problem such as a large number of learners in the dataset Assist15 and few average answer records.

D. Ablation experiments

To verify the effectiveness of each module setting in the MPHINE-KT model proposed in this paper, ablation experiments were conducted on this model. The experiments were also carried out on four datasets: ASSIST09, ASSIST15, Algebra05, and Bridge06. This part mainly ablated the following three modules: (1) the learner behavior graph module; (2)

the SOM clustering module; (3) the learner heterogeneous information network module. The descriptions of the models after ablating these modules are as follows:

(1) MPHINE-RLBG (Replace Learner Behavior Graph): The learner behavior graph module, which is used to extract learners' learning experience, is removed. Instead, the exercise representations of the learner's answer behavior sequences are directly accumulated and then put into an MLP layer to extract the learners' learning experience.

(2) MPHINE-RSOM (Replace SOM clustering): The SOM clustering module for the representation of the learner behavior graph is removed and replaced with the K-Means clustering algorithm. An appropriate number of clusters k is selected to aggregate similar learners' learning experience.

(3) MPHINE-RLHIN (Remove Learners' Heterogeneous Information Networks): The learner heterogeneous information network module and the subsequent embedding extraction module are removed. The vectors extracted from the learner behavior graph embeddings are concatenated with the exercise vectors and the learners' responses and finally input into the GRU network to predict the learners' learning states.

Except for the changes mentioned above, the implementation of other parts in the ablation experiments remains consistent with the model described in the model section of this paper. The experimental results are shown in the TABLE V. By comparing the AUC and ACC metrics of the ablated models with those of the MPHINE-KT, the results are as follows: (1) Among all the models, the MPHINE-KT model achieved the best performance in terms of AUC and ACC metrics. This is strong evidence that the model can effectively mine learners' learning experience, learning abilities, and the interactions among learners. It overcomes the issue of sparse input data in knowledge tracing tasks, which in turn improves the model's prediction performance. (2) When the learner behavior graph module was replaced, the AUC and ACC performance on multiple datasets dropped significantly. This clearly shows that constructing a learner behavior graph and extracting its graph representations can help obtain more accurate insights into learners' learning experience. This allows for more precise clustering and construction of the learner heterogeneous information network. (3) The optimal value of k for K-Means in the MPHINE-RSOM model was selected based on experimental performance. However, the clustering effect was still not as

TABLE V
COMPARISON OF ABLATION EXPERIMENT RESULTS

Model	AUC				ACC			
	Assist09	Assist15	Algebra05	Bridge06	Assist09	Assist15	Algebra05	Bridge06
MPHINE-RLBG	0.8388	0.7187	0.9149	0.8012	0.7766	0.7354	0.8489	0.8356
MPHINE-RSOM	0.8443	0.7132	0.9025	0.8084	0.7883	0.7352	0.8563	0.8480
MPHINE-RLHIN	0.8102	0.7121	0.8962	0.7928	0.7484	0.7232	0.8321	0.8362
MPHINE-KT	0.8501	0.7364	0.9229	0.8167	0.7876	0.7584	0.8637	0.8505

good as that of using SOM for representing learners' learning experience. A likely reason is that the performance of the K-Means algorithm is highly sensitive to the parameter k , while SOM clustering can better maintain the structure of the input data. (4)After removing the learner heterogeneous information network in the MPHINE-RLHIN model, the performance also decreased. The average decline in ACC and AUC was the largest among the three ablated models. This result demonstrates the effectiveness of integrating learning ability features and constructing the learner heterogeneous information network. In the dataset, learners' knowledge states are affected by both their learning abilities and learning experience. Moreover, there are correlations among learners in terms of learning abilities and learning experience. The learner heterogeneous information network can capture such information, reduce the sparsity of knowledge tracing inputs, and improve the model's prediction performance.

E. Visual Analysis of Learner Predictions

In order to illustrate the accuracy of the MPHINE-KT prediction, experiment selected the question-answering record of a learner in the Assist09 dataset, and intercept some of its outputs in the MPHINE-KT for visualization, as shown in Fig. 6. The horizontal coordinates in the picture represent the learner's question-answering time series, and a tuple of two numbers is used to represent an interaction record. The depth of the color of the squares in the picture represents the predicted degree of learners' knowledge mastery, with the darker color indicating better mastery and the lighter color indicating worse mastery. In practice, if a learner answers a question incorrectly consecutively, it means that the learner's mastery of the question is poor and the probability of answering the question correctly in the future is small. For example, in the figure, the learner answered question 50 consecutively. Although there were cases of wrong answers, the learner's knowledge mastery of the question subsequently turned out to be better after considering the learner's individual integrated learning attributes. Therefore, the visualization of the learner prediction process shows that when the learner's answer changes, the model is able to detect changes in the mastery of the concepts, understand the learner's knowledge level, and thus personalize the guided learning service for the learner.

VI. DISCUSSION

Experimental results show that MPHINE-KT outperforms the given baseline models in AUC and ACC. This shows it can effectively extract the representation of learners' answering behavior sequences in knowledge tracing, enhancing the prediction of learners' future performance. Compared with

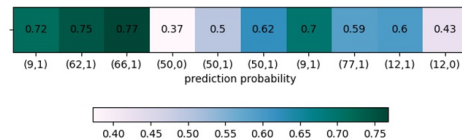


Fig. 6. Visualization of Learner Learning States Predicted by MPHINE-KT

DKT on the given dataset, MPHINE-KT improves AUC and ACC to varying extents. By leveraging a heterogeneous information network, it comprehensively analyzes how learners' learning experience, and individual and general learning abilities, influence knowledge tracing predictions. Incorporating learners' individual attributes alleviates input data sparsity, resulting in better time-series prediction performance. The ablation experiments compared the models with important modules ablated to MPHINE-KT and found that the performance of these ablated models decreased to varying degrees. This proves the effectiveness of each module in MPHINE-KT. The experiments conducted verify the design intention of MPHINE-KT: By exploring learners' individual attributes and heterogeneous information in the answering sequences, it can effectively quantify learners' characteristics, help alleviate the problem of sparse input data in knowledge tracing, thus improving the model's performance and providing support for key educational aspects such as personalized education.

Learners are a very important part of the knowledge tracing task. The learners' answering behavior sequences contain rich learner representations. Effectively utilizing these representations can alleviate the problem of input data sparsity in the knowledge tracing task. Most of the existing work addresses the sparsity problem from the perspectives of exercises and knowledge points, but does not fully utilize the individual attribute features in learners' answering behavior characteristics. Some of the related research transforms the learners' answering behavior sequence data into homogeneous graph structure data, which makes it difficult to explore the connections between different types of attributes. The model proposed in this paper constructs a learner heterogeneous information network to capture the heterogeneous information in the sequence data, thereby improving the performance of the model.

MPHINE-KT can empower online education systems and enhance the efficiency of exercise recommendation. The primary objective of an online exercise recommendation system is to offer students effective learning plans. It can provide the mastered knowledge states of students and the representations of students that integrate features of learning abilities and experiences for downstream exercise recommendation tasks,

thereby tailoring learning plans for different students according to their learning characteristics.

There are still many learners' behavior characteristics that can be extracted from the data of the knowledge tracing task. For example, learners' learning styles can affect their knowledge states, and learners with similar learning styles may have similar knowledge states. The extraction of learners' characteristics in this paper is not comprehensive enough, and appropriate methods will be further explored in future work to improve the accuracy of the knowledge tracing task in evaluating learners' learning states.

VII. CONCLUSION

This paper carries out research on personalized teaching problems in smart education scenarios, and take learners' question-answering behavior data as the research entry point to explore the knowledge tracing model in depth. This paper solves the problems of existing knowledge tracing models that lack generality, ignore the correlation between learners' question-answering behavior and learners' learning experience, and the influence of learners' individual attribute characteristics on learners' performance prediction. A knowledge tracing model called MPHINE-KT is proposed, which firstly constructs and analyzes the learner behavior graph based on the learner's learning interaction records, and extracts the learners' learning experience attributes by SOM clustering of the learned graph representations; and then calculates the learner's learning ability to construct the learner heterogeneous information network and learn to characterize learner nodes from node-level and meta-path-level attention mechanisms, respectively. Finally, the learner node representations are put into the GRU model as additional inputs for predicting the learner's learning status. The experimental results prove that the proposed model has better performance compared to existing popular knowledge tracing models, which also confirms the effectiveness of employing learning experience recognition neural network and constructing learner heterogeneous information network.

The work in this paper can be further expanded. For instance, the performance of the model can be further enhanced, and the forgetting situation of learners can be more comprehensively considered in the knowledge tracing model.

REFERENCES

- [1] U. Barbieri, E. Marsico, L. Picci, R. Di Fuccio, and F. P. Cassese, "The future of education: Personalized learning through adaptive intelligent tutoring systems with natural language and deep learning," *Italian Journal of Health Education, Sports and Inclusive Didactics*, vol. 8, no. 3, 2024, doi: 10.32043/gsd.v8i3.1179.
- [2] H. A. Alrakhawi, N. Jamiat, and S. S. Abu-Naser, "Intelligent tutoring systems in education: A systematic review of usage, tools, effects and evaluation," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 4, pp. 1205–1226, 2023.
- [3] A. Królak and P. Zajac, "Analysis of the accessibility of selected massive open online courses (MOOCs) for users with disabilities," *Universal Access in the Information Society*, 2024, doi: 10.1007/s10209-022-00927-2.
- [4] S. Haggard, S. Brown, R. Mills, A. Tait, and S. Warburton, *The Maturing of the MOOC: Literature Review of Massive Open Online Courses and Other Forms of Online Distance Learning*. London, U.K.: Department for Business, Innovation and Skills, 2013.
- [5] M. M. Nujid and D. A. Tholibon, "A review of engagement strategies for massive open online courses," *International Journal of Evaluation and Research in Education*, vol. 13, no. 5, 2024.
- [6] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Modeling and User-Adapted Interaction*, vol. 4, no. 4, pp. 253–278, 1994, doi: 10.1007/BF01099821.
- [7] X. Song, J. Li, T. Cai, S. Yang, T. Yang, and C. Liu, "A survey on deep learning based knowledge tracing," *Knowledge-Based Systems*, 2022, Knowledge-Based Systems, vol. 258, no. 110036, 2022, doi: 10.1016/j.knsys.2022.110036.
- [8] N. Zhai and X. Ma, "Automated writing evaluation (AWE) feedback: A systematic investigation of college students' acceptance," *Computer Assisted Language Learning*, 2022, doi: 10.1080/09588221.2021.1897019.
- [9] X. Chen and H. Deng, "Research on personalized recommendation methods for online video learning resources," *Applied Sciences*, vol. 11, no. 2, Art. no. 804, 2021, doi: 10.3390/app11020804.
- [10] M. I. Ahmed Al-Mashhadani, T. H. H. Aldhyani, M. H. Al-Adhaileh, A. M. Bamhdi, M. Y. Alzahrani, F. W. Alsaade, and H. Alkahtani, "Human-animal affective robot touch classification using deep neural network," *Computer Systems Science and Engineering*, vol. 38, no. 1, 2021.
- [11] Z. Wu, T. He, C. Mao, and C. Huang, "Exam paper generation based on performance prediction of student group," *Information Sciences*, vol. 532, pp. 72–90, 2020, doi: 10.1016/j.ins.2020.04.043.
- [12] J. Kasurinen and U. Nikula, "Estimating programming knowledge with Bayesian knowledge tracing," *ACM SIGCSE Bulletin*, vol. 41, no. 3, pp. 313–317, 2009, doi: 10.1145/1595496.1562972.
- [13] C. Piech, J. Spencer, J. Huang, S. Ganguli, M. Sahami, L. Guibas, and J. Sohl-Dickstein, "Deep knowledge tracing," in *Advances in Neural Information Processing Systems*, vol. 28, pp. 505–513, 2015.
- [14] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, Art. no. 132306, 2020, doi: 10.1016/j.physd.2019.132306.
- [15] D. Liu, Y. Zhang, J. Zhang, Q. Li, C. Zhang, and Y. Yin, "Multiple features fusion attention mechanism enhanced deep knowledge tracing for student performance prediction," *IEEE Access*, vol. 8, pp. 194894–194903, 2020, doi: 10.1109/ACCESS.2020.3033200.
- [16] Y. Sun, L. Wang, Q. Xie, Y. Dong, and X. Lin, "Online programming education modeling and knowledge tracing," in *Proc. International Conference on Knowledge Science, Engineering and Management*, 2020, doi: 10.1007/978-3-030-55130-8_23.
- [17] N. Zhang, Y. Du, K. Deng, L. Li, J. Shen, and G. Sun, "Attention-based knowledge tracing with heterogeneous information network embedding," in *Proc. International Conference on Knowledge Science, Engineering and Management*, 2020, doi: 10.1007/978-3-030-55130-8_9.
- [18] Y. Lu, P. Chen, Y. Pian, and V. W. Zheng, "CMKT: Concept map driven knowledge tracing," *IEEE Transactions on Learning Technologies*, vol. 15, no. 4, pp. 467–480, Aug. 2022, doi: 10.1109/TLT.2022.3196355.
- [19] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, Jan. 2021, doi: 10.1109/TNNLS.2020.2978386.
- [20] C. Shi, Y. Li, J. Zhang, Y. Sun, and P. S. Yu, "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 17–37, Jan. 2017, doi: 10.1109/TKDE.2016.2598561.
- [21] R. Bing, G. Yuan, M. Zhu, F. Meng, H. Ma, and S. Qiao, "Heterogeneous graph neural networks analysis: A survey of techniques, evaluations and applications," *Artificial Intelligence Review*, vol. 56, no. 8, pp. 8003–8042, 2023, doi: 10.1007/s10462-022-10375-2.
- [22] T.-Y. Fu, W.-C. Lee, and Z. Lei, "HIN2Vec: Explore meta-paths in heterogeneous information networks for representation learning," in *Proc. ACM International Conference on Information and Knowledge Management*, 2017, doi: 10.1145/3132847.3132953.
- [23] C. Zhang, A. Swami, and N. V. Chawla, "SHNE: Representation learning for semantic-associated heterogeneous networks," in *Proc. 12th ACM International Conference on Web Search and Data Mining*, 2019, doi: 10.1145/3289600.3291001.
- [24] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *Proc. The World Wide Web Conference*, 2019, pp. 2022–2032, doi: 10.1145/3308558.3313562.
- [25] X. Fu, J. Zhang, Z. Meng, and I. King, "MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proc. The Web Conference 2020*, 2020, pp. 2331–2341, doi: 10.1145/3366423.3380297.
- [26] B. Hu, Y. Fang, and C. Shi, "Adversarial learning on heterogeneous information networks," in *Proc. 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, doi: 10.1145/3292500.3330970.

- [27] K. Zhao, T. Bai, B. Wu, B. Wang, Y. Zhang, Y. Yang, and J. Y. Nie, "Deep adversarial completion for sparse heterogeneous information network embedding," in *Proc. The Web Conference 2020*, 2020, doi: 10.1145/3366423.3380134.
- [28] S. R. Eddy, "Hidden Markov models," *Current Opinion in Structural Biology*, vol. 6, no. 3, pp. 361–365, 1996, doi: 10.1016/S0959-440X(96)80056-X.
- [29] W. J. Hawkins and N. T. Heffernan, "Using similarity to the previous problem to improve Bayesian knowledge tracing," in *Proc. Educational Data Mining Workshops*, 2014.
- [30] D. Agarwal, R. S. Baker, and A. Muraleedharan, "Dynamic knowledge tracing through data driven recency weights," in *Proc. 13th International Conference on Educational Data Mining*, 2020.
- [31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [32] J. Zhang, X. Shi, I. King, and D. Y. Yeung, "Dynamic key-value memory networks for knowledge tracing," in *Proc. 26th International Conference on World Wide Web*, 2017, pp. 765–774, doi: 10.1145/3038912.3052580.
- [33] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, A. P. Badia, K. M. Hermann, Y. Zwols, G. Ostrovski, A. Cain, H. King, C. Summerfield, P. Blunsom, K. Kavukcuoglu, and D. Hassabis, "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, pp. 471–476, 2016, doi: 10.1038/nature20101.
- [34] C. K. Yeung, "Deep-IRT: Make deep learning based knowledge tracing explainable using item response theory," *arXiv preprint arXiv:1904.11738*, 2019.
- [35] G. Rasch, *Probabilistic Models for Some Intelligence and Attainment Tests*. Chicago, IL, USA: MESA Press, 1993.
- [36] S. Pandey and G. Karypis, "A self-attentive model for knowledge tracing," *arXiv preprint arXiv:1907.06837*, 2019.
- [37] A. Ghosh, N. Heffernan, and A. S. Lan, "Context-aware attentive knowledge tracing," in *Proc. 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020, doi: 10.1145/3394486.3403282.
- [38] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, vol. 27, 2014, pp. 3104–3112.
- [39] H. Nakagawa, Y. Iwasawa, and Y. Matsuo, "Graph-based knowledge tracing: Modeling student proficiency using graph neural network," in *Proc. IEEE/WIC/ACM International Conference on Web Intelligence*, 2019, doi: 10.1145/3350546.3352513.
- [40] Y. Yang, J. Shen, Y. Qu, Y. Liu, K. Wang, Y. Zhu, W. Zhang, and Y. Yu, "GIKT: A graph-based interaction model for knowledge tracing," in *Proc. Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2020, doi: 10.1007/978-3-030-67658-2_18.
- [41] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. International Conference on Learning Representations*, 2017.
- [42] X. Song, J. Li, Y. Tang, T. Zhao, Y. Chen, and Z. Guan, "JKT: A joint graph convolutional network based deep knowledge tracing," *Information Sciences*, vol. 580, pp. 510–523, 2021, doi: 10.1016/j.ins.2021.08.100.
- [43] W. Gan, Y. Sun, and Y. Sun, "Knowledge structure enhanced graph representation learning model for attentive knowledge tracing," *International Journal of Intelligent Systems*, 2022, doi: 10.1002/int.22763.
- [44] J. Sun, S. Du, Z. Liu, F. Yu, S. Liu, and X. Shen, "Weighted heterogeneous graph-based three-view contrastive learning for knowledge tracing in personalized e-learning systems," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 2838–2847, 2024, doi: 10.1109/TCE.2023.3293953.
- [45] J. Liu, Z. Guan, X. Wang, and Z. Yang, "Heterogeneous graph knowledge tracing with novel contrastive view and meta-path contexts," *Expert Systems with Applications*, vol. 295, Art. no. 128840, 2026, doi: 10.1016/j.eswa.2025.128840.
- [46] Y. Wu, H. Zhu, C. Wang, F. Song, H. Zhu, Y. Chen, Q. Zheng, and F. Tian, "Programming knowledge tracing based on heterogeneous graph representation," *Knowledge-Based Systems*, vol. 300, Art. no. 112161, 2024.
- [47] Q. Ni, T. Wei, J. Zhao, L. He, and C. Zheng, "HHSKT: A learner-question interactions based heterogeneous graph neural network model for knowledge tracing," *Expert Systems with Applications*, vol. 215, Art. no. 119334, 2023, doi: 10.1016/j.eswa.2022.119334.
- [48] C. Jiang, W. Gan, G. Su, Y. Sun, and Y. Sun, "Improving knowledge tracing through embedding based on metapath," in *Proc. International Conference on Computers in Education*, 2021.
- [49] T. Wu and Q. Ling, "Self-supervised heterogeneous hypergraph network for knowledge tracing," *Information Sciences*, vol. 624, pp. 200–216, 2023, doi: 10.1016/j.ins.2022.12.075.
- [50] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990, doi: 10.1109/5.58325.
- [51] M. Feng, N. Heffernan, and K. Koedinger, "Addressing the assessment challenge with an online system that tutors as it assesses," *User Modeling and User-Adapted Interaction*, vol. 19, no. 3, pp. 243–266, 2009, doi: 10.1007/s11257-009-9063-7.
- [52] C. K. Yeung and D. Y. Yeung, "Addressing two problems in deep knowledge tracing via prediction-consistent regularization," in *Proc. 5th Annual ACM Conference on Learning at Scale*, 2018, doi: 10.1145/3231644.3231647.
- [53] C. Cui, Y. Yao, C. Zhang, H. Ma, Y. Ma, Z. Ren, C. Zhang, and J. Ko, "DGEKT: A dual graph ensemble learning method for knowledge tracing," *ACM Transactions on Information Systems*, vol. 42, no. 3, pp. 1–24, 2024, doi: 10.1145/3638350.