International Journal of Data Warehousing and Mining
Volume 20 • Issue 1

GTFN:
Knowledge Tracing Model Based on Graph Temporal Fusion Networks

Meng Huang, Xi ‘an Technological University, China*
Ting Wei, Xi ‘an Technological University, China

ABSTRACT

With the development of smart education, gaining insights into students’ understanding during the learning process is crucial in teaching. However, traditional knowledge tracking methods face challenges in capturing the intricate relationships between problems and knowledge points, as well as students’ temporal learning changes. Therefore, we design a knowledge tracking model based on a graph temporal fusion network. Firstly, we construct the structure of the question and knowledge skill graph. Then, we design a knowledge graph encoder layer to capture the complex relationships between questions and knowledge skills. Next, we apply a sequential information extraction layer to dynamically model the outputs of each layer in the upper network over time, capturing students’ knowledge changes at different time steps. Finally, we use a dynamic attention aggregation network to learn node information at different levels and time sequences. Experimental results on three datasets demonstrate the effectiveness of our method.

KEYWORDS

Attention Network, Gated Recurrent Network, Graph Attention Network, Knowledge Tracing

In the modern educational environment (Park & Kwon, 2023), there is a growing demand for personalized learning and intelligent-assisted education. Understanding students’ mastery levels during the learning process is crucial for providing tailored educational support. Traditional assessment methods often struggle to accurately capture students’ learning processes. KT, as an assessment method based on student behavior, offers a more comprehensive understanding of students’ comprehension levels by analyzing their learning activities.

KT involves predicting a student’s mastery of specific concepts based on their learning activity history. This not only provides targeted teaching recommendations for educators but also offers personalized learning paths for students. In the past, KT models primarily utilized Bayesian network methods. However, with the widespread adoption of DL techniques, KT methods based on DL have become increasingly prevalent.

DOI: 10.4018/IJDWM.345406

*Corresponding Author

Corbett and Anderson (2005) first introduce the incorporation of Bayesian Knowledge Tracing into intelligent teaching. The core of this approach is based on an HMM probabilistic model with a focus on time-series data. It tracks and analyzes the probability of students mastering a knowledge point at the next time step. Agarwal et al. (2020) propose the MS-BKT to address issues in the BKT model, such as the constant learning rate and only two knowledge states. The MS-BKT model extends the knowledge states to 21, offering a more effective assessment of students' learning states. In recent years, DL methods, known for their powerful feature extraction capabilities and the ability to operate without the need for manually-labeled data, have garnered widespread attention among researchers. Piech et al. (2015) introduce the DKT model. Although this method effectively alleviates many issues associated with traditional BKT models, the lack of interpretability and the absence of learned features in the input and output mechanisms of DL models constrain the practical application of KT models in teaching. Following the remarkable success of GNN in various fields, Nakagawa et al. (2019) introduce the GKT based on GNN (Zhou et al., 2018). It formulates a conceptual graph structure, treating the KT task as a node-level classification problem. Nevertheless, it inadequately incorporates information about the learner's evolving states across different time steps. Pandey & Srivastava (2020) propose RKT, incorporating a context-aware self-attention network layer that integrates exercise relationship character and student performance data. In general, many methods can construct graph structures or use attention networks to learn students' learning state information effectively. However, these methods tend to capture students' learning states from a singular perspective, such as using GCN or attention networks to learn high-order relationships between problems and knowledge points. This limitation makes it challenging to capture the complex relationships between problems and knowledge points effectively. Additionally, there is a scarcity of research considering the evolving information about students' learning states over time.

Based on this, we design the KT Model based on GTFN. Firstly, we construct the structure of the question and knowledge skill graph. Then, we employ the GAT to capture the complex relationships between problems and knowledge points (Velickovic et al., 2017; Yang et al., 2023). A sequential dynamic modeling approach is introduced, utilizing the GRU to perform temporal dynamic modeling on the output of the multi-layered GAT network (Han et al., 2022), effectively capturing changes in students' knowledge at different time steps. Finally, a Dynamic Attention Aggregation Network is proposed, using the GAT network output as the query matrix for the attention network, effectively integrating node information at different levels and time sequences. To summarize, our principal contributions can be outlined as follows:

1. Graph Temporal Fusion Method: We utilize the GAT to capture the complex relationships between problems and knowledge points. This enhances effective understanding of students' learning processes.
2. Temporal Dynamic Modeling: We employ a multi-layered GRU network for sequential dynamic modeling of GAT network outputs. This captures changes in students' knowledge at different time steps, strengthening the model's capability for temporal dynamic modeling of learning trajectories.
3. Dynamic Attention Aggregation Network: We introduce a dynamic attention mechanism that comprehensively considers GAT and GRU network outputs, integrating node information at different levels and time sequences. This improves the model's adaptability to the complexity of students' learning trajectories.

The remaining sections of this paper are organized as follows: We introduce related work; define the problem; present the DTFN model; cover experiments and evaluations; discuss the findings; and summarize the entire document.

RELATED WORK

KT Models Based on Bayesian

Based on the BKT (Corbett & Anderson, 2005), which is a commonly used approach in establishing models for student learning sequences, an HMM is employed to treat a student's knowledge state as a latent variable. Nevertheless, this methodology is oriented around KCs, where KC serves as a universal term encompassing knowledge points, concepts, skills, or items. Consequently, all students share a common set of parameters for a given KC, leading to a situation in which students at an intermediate or higher proficiency level persistently receive a substantial volume of recommended exercises, even after mastering a specific KC. This, in turn, necessitates the completion of redundant practice questions. To address this issue, many scholars are extending the BKT model from different perspectives to enhance its practicality and accuracy. Pardos and Heffernan (2010) propose a learning model whereby different students have different initial background knowledge probabilities, enabling more accurate estimates of when a student masters a KC.

As most of the extended BKT models are based on HMM, they assume a constant learning rate after answering questions, making it difficult to balance recent and historical exercise data for students, which is not in line with the objective reality. Agarwal et al. (2020) propose the MS-BKT model, replacing the fixed learning rate with a recent rate weight based on a student's overall situation. This method captures the student's progress through data rather than attempting to assume continuous learning. The model extends the knowledge states from the typical two states to 21 states, steadily updating estimates over time. This better captures the complexity of correct and incorrect sequences. Overall, Bayesian network-based methods have relatively simple model constructions but possess powerful interpretability.

Knowledge Tracing Models Based on DL

Following the advancement of DL technology, an increasing number of DL models have been thoroughly researched and introduced into KT tasks. Piech et al. (2015) propose the DKT method, which is the first to apply DL to KT tasks. It captures more complex student knowledge states without the need for explicit encoding of the knowledge domain. Su et al. (2021) introduce TC-MIRT, constructing time-enhanced and concept-enhanced network components. This enables the model to perform trend predictions and generate interpretable parameters within each specific knowledge domain. Therefore, Minn et al. (2018) propose DKT-DSC, combining k-means clustering with Euclidean distance. By considering students' abilities and practice skill personalized input vectors, DKT-DSC captures the evolution of student capabilities over time, periodically and dynamically assigning students to different groups with similar abilities. Lasheng (2020) introduces the SAEN model, which integrates heterogeneous features of students with relevant information on forgetting behaviors.

Due to the DKT model representing the mastery of students over KC with hidden states, it cannot provide detailed output regarding students' mastery levels for each KC. Yang S. et al. (2020) combine the forgetting curve theory, which posits that human memory declines over time, and propose the CKT model. This model uses LSTM to capture long-term features and leverages 3D Conv to enhance the short-term effects of recent exercises, effectively modeling the forgetting curve in students' learning processes. Nakagawa et al. (2019), for the first time, apply GNN to KT, proposing the GKT method. This method enhances prediction accuracy without relying on any additional information. However, this method only uses KC as input, overlooking issues such as multiple KCs and the impact between exercises. Yang Y. et al. (2020) introduce the GIKT model, utilizing GCN networks to aggregate exercise-KC embedding features learned from high-order relationships. It directly incorporates exercise embeddings with corresponding answer embeddings as inputs for the KT model.

Although GCN performs well in aggregating neighborhood features on graph structures, it treats each node feature equally when aggregating neighborhood features, which is unfair. Ghosh et al. (2020) propose the AKT model, which uses a novel monotonic attention network to connect learners' future answers to assessment questions with their past answers. This approach captures individual differences under the same question concept without the need for excessive parameters. However, the attention layers are shallow, making it challenging to learn relationships between features effectively. Tu et al. (2023) introduce DGKT to address two major shortcomings in knowledge tracing research. It proposes static embeddings and dynamic embeddings, representing stable attributes and time-varying properties of students, questions, and concepts respectively, and utilizes graph self-supervised learning to enrich static embeddings of questions and concepts. To consider the varying importance of different questions for students, Liu et al. (2022) introduce QDEKT, which comprehensively considers the difficulty of questions and knowledge concepts. They incorporate students' real and predicted responses to assess students' knowledge acquisition capabilities and dynamically update their knowledge states. Introducing additional features can effectively enhance predictive capabilities. Yang et al. (2022) propose SHDKT, introducing skill concurrency graphs and a hierarchical skill representation module to explore sequential and co-occurrence relationships between skills. They supplement students' corresponding information into the question and knowledge skill graphs using a multi-layer GNN network to learn hierarchical representations of skills, where each layer represents a skill level. Extracting information representations at different levels has a certain effect. However, it does not account for the dynamic changes in students' learning states. Cui et al. (2024) propose DGEKT, which establishes a dual graph structure of student learning interactions, capturing exercise-concept associations and interaction transitions. Additionally, it introduces online knowledge distillation to form a stronger ensemble teacher model, enhancing modeling capability.

In summary, current approaches vary in their proposals, some suggesting the use of graph algorithms to learn node feature information from the structures of question and knowledge skill graphs, while others augment features to better represent students' learning states. However, few methods consider incorporating the evolving learning state information of students while utilizing graph-based algorithms. Considering this, we propose the GTFN method. Firstly, it constructs a graph of questions and knowledge skills, then employs a GAT to learn node feature representations from the graph. Utilizing a GRU network capable of capturing students' learning states, it captures students' knowledge changes at different time steps across multiple GAT network layers, achieving temporal dynamic modeling of learning trajectories. Overall, the model effectively captures both node feature information within the graph and the changing information of students' learning states.

PRELIMINARY

Symbol Annotations

See Tables 1 and 2.

Knowledge Tracing

Consider a time series representing a student's learning process $S = \left\{\left(q_1, a_1\right), \left(q_2, a_2\right), \ldots, \left(q_t, a_t\right)\right\}$, where q_t is the t-th question, and a_t is the student's response to the question (binary value). The objective of KT is to predict the student's response $P\left(a_{t+1} \middle| S\right)$ to a given question q_{t+1} at the $t+1$ time step.

Table 1. Symbol Annotations for the GTFN Model

Names	Details
q_t	The t -th question
a_t	Accuracy of student responses to questions
Q	Set of Questions
K	Set of Knowledge Points
G	Question-Knowledge Graph
h_v	Initial Feature Representation of Node v
e_{ij}^{l}	Attention Weight Coefficients
h_i^{l+1}	Representation of Node i
$\alpha_{ij}^{l,k}$	Attention Coefficients for the k -th Subspace
$h_v^{GRU_l}$	GRU Processing on GAT Layer l Output
h_v^{final}	The Final Node Representation

Question and Knowledge Skill Graph

Assuming there is a set of questions $Q = \left\{q_1, q_2, \ldots, q_M\right\}$ and a set of knowledge points $K = \left\{k_1, k_2, \ldots, k_N\right\}$, we construct a graph $G = (V, E)$ where the node set V includes questions or knowledge points. The edge set E represents the relationships between questions and knowledge points, which can be determined based on the correlation or dependency between questions and knowledge points.

METHOD

Overview of the GTFN Model

The KT Model based on GTFN comprises several components. As shown in Figure 1, by constructing a graph structure for questions and knowledge points, the model then utilizes a Knowledge Graph Encoder layer to introduce GAT to capture the intricate relationships. Subsequently, in the Sequential Information Integrator layer, a multi-layer GRU is employed to model the temporal evolution of knowledge dynamically, capturing changes in a student's understanding at different time steps. Finally, through the Dynamic Attention Aggregation Network, the model effectively integrates the outputs of GAT and GRU, considering node information at different levels and time sequences.

Table 2. The Table of Abbreviations

Abbr.	Full Name
AKT	Attentive Knowledge Tracing
BKT	Bayesian Knowledge Tracing Model
CKT	Convolutional Knowledge Tracing
DAAN	Dynamic Attention Aggregation Network
DKT-DSC	DKT With Dynamic Student Classification
DL	Deep Learning
DGKT	A Graph-Based Dynamic Interactive Knowledge Tracing Method
DGEKT	A Dual Graph Ensemble Learning Method for Knowledge Tracing
GAT	Graph Attention Network
GCN	Graph Convolutional Networks
GIKT	Graph-Based Interaction Model for Knowledge Tracing
GKT	Graph Knowledge Tracing Model
GKT	Graph Knowledge Tracing
GNN	Graph Neural Networks
GRU	Gated Recurrent Unit
GTFN	Knowledge Tracing Model based on Graph Temporal Fusion Networks
HMM	Hidden Markov Model
KCs	Knowledge Components
KT	Knowledge Tracing
MS-BKT	Multistate-Bayesian Knowledge Tracing Model
QDEKT	Knowledge Tracing with Multi-Feature Fusion and Question Difficulty
RKT	Relation Knowledge Tracing
SAEN	Stacked Auto-Encoder Network
TC-MIRT	Time-And-Concept Enhanced Deep MIRT

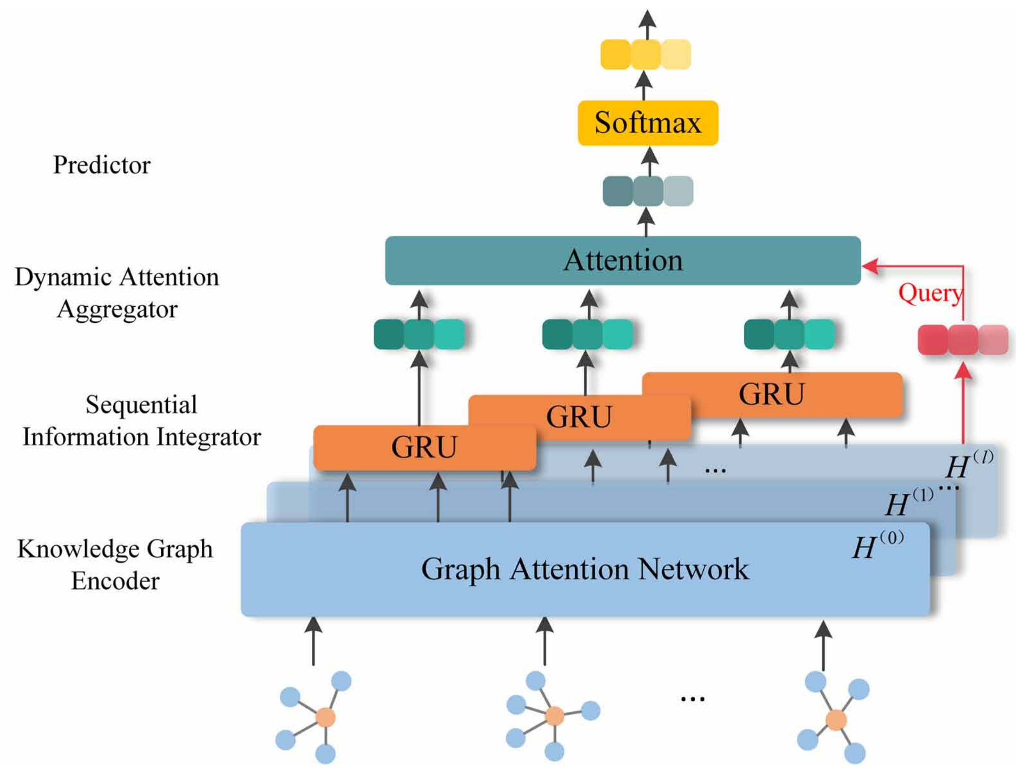
Knowledge Graph Encoder

The GAT is a type of GNN network designed specifically to handle complex relationships between nodes in a graph structure. In KT tasks, there may be various types of associations between questions and knowledge points, and this network can effectively capture these intricate relationships through an adaptive attention mechanism.

For each node $v \in V$, the Embedding method is employed to obtain the initial feature representation h_v (Cui et al., 2017). For each layer l, the attention weight coefficients e_{ij}^{l}, attention coefficients α_{ij}^{l}, and the weighted aggregation features h_i^{l+1} for the neighboring nodes of node i are computed as follows:

$$e_{ij}^{l} = \mathrm{LeakyReLU}\left(\mathbf{a}^{l}\left(\mathbf{W}^{l}h_i^{l} \middle\| \mathbf{W}^{l}h_j^{l}\right)\right) \tag{1}$$

Figure 1. Overall Framework of the GTFN Model



$$\alpha_{ij}^{l} = \frac{\exp\left(e_{ij}^{l}\right)}{\sum_{k\mathcal{N}_i} \exp\left(e_{ik}^{l}\right)} \tag{2}$$

$$h_i^{l+1} = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}^{l}\left(\mathbf{W}^{l} h_j^{l}\right)\right) \tag{3}$$

where $\mathcal{N}_i$ represents the set of adjacent nodes for node i, and σ is the activation function.

Subsequently, a multi-head attention network is employed to compute multiple sets of attention weights in parallel. Finally, these sets are concatenated together:

$$h_i^{l+1} = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}^{l,k}\left(\mathbf{W}^{l,k} h_j^{l}\right)\right) \tag{4}$$

where $\alpha_{ij}^{l,k}$ is the attention coefficient for the k-th subspace.

Sequential Information Integrator

GRU is a type of RNN structure suitable for sequential data, effectively capturing temporal information within input sequences. As the output of each layer in the GAT network can be considered as the evolution of node information at different time steps, this method is employed to model these dynamics. By processing the network outputs at each layer, GTFN can dynamically learn representations of nodes at different hierarchical levels. The gating mechanism assists the model in selectively updating and forgetting information.

The output $[h_v^1, h_v^2, ..., h_v^L]$ of each layer l in the GAT network serves as the input sequence for the GRU, where L is the layer number. Subsequently, the GRU network is used to process the input sequences for each node, resulting in the output $h_v^{GRU_l}$ for each layer's information:

$$h_v^{GRU_l} = GRU([h_v^1, h_v^2, ..., h_v^L]) \tag{5}$$

Dynamic Attention Aggregation Network

To adjust the focus of attention dynamically based on the specific context of questions and knowledge points, a dynamic and flexible attention mechanism is designed. This helps improve the model's generalization, allowing it to perform well on various combinations of questions and knowledge points.

By using the output of the last layer of the GAT network as the query matrix, the model can globally attend to the features of nodes in the graph structure. This aids in better understanding of the importance of nodes throughout the entire graph. The key-value matrix of the attention network is formed by the multi-layer outputs of the GRU network, as it provides information about nodes at different levels. This includes features from different abstract levels, ranging from lower to higher layers. Concatenating these matrices enables the model to consider information from different hierarchical levels comprehensively:

$$Q_v = W^Q h_v^L \tag{6}$$

$$h_v^{GRU} = [h_v^{GRU_1}, h_v^{GRU_2}, ..., h_v^{GRU_L}] \tag{7}$$

$$K_v = W^K h_v^{GRU} \tag{8}$$

$$V_v = W^V h_v^{GRU} \tag{9}$$

where W^Q and W^K represent weight coefficients, respectively.

Finally, by calculating the attention weights α_v between node v and other nodes, a weighted aggregation is applied to the key-value matrix, resulting in the ultimate representation of node v:

$$\alpha_v = \mathrm{softmax}(Q_v K_v^T) \tag{10}$$

$$h_v^{final} = \sum_i \alpha_{vi} V_{vi}. \tag{11}$$

Fully Connected Layer

The node representations aggregated through the attention network are fed into the SoftMax method, yielding the prediction result p:

$$p = \mathrm{softmax}(W^t h^{final} + b^t) \tag{12}$$

Optimization

The objective is to minimize the difference between the predicted probability distribution and the actual probability distribution, hence the utilization of the cross-entropy loss function (Li et al., 2020). This loss measure penalizes larger disparities, guiding the iterative adjustment of model parameters during training.

$$\mathcal{L} = -\sum_t (y_t \log p_i + (1 - y_i)\log(1 - p_i)) \tag{13}$$

Algorithm 1. GTFN

1: **Initialize:**
Weights, bias, graph G
Begin:
2: **For** epoch in ep **do**
3: Get the embedding vectors h_v
4: Capture the intricate relationships h_i^l
5: Capture the temporal information h_v^{GRU}
6: Capture learning state information h_v^{final}
7: Obtain prediction results p_i
8: Update parameters
9: **End For**

EXPERIMENT

We compare GTFN with other methods on three public datasets to assess prediction performance and answer three questions:

1. How does the performance of the GTFN method compare to state-of-the-art methods?
2. What is the impact of key model designs of GTFN on experimental results?
3. How does the setting of hyperparameters affect the experimental results?

Dataset

ASSIST2012[2]: This dataset is similarly collected from the platform utilized for ASSIST2009. It has undergone processing procedures identical to those applied to the ASSIST2009 dataset.

Table 3. Datasets

	ASSIST2009	ASSIST2012	Algebra2006
#students	3,043	28758	1,087
#questions	18,267	52,136	132,473
#Concepts	117	256	543
#Records	265,734	2,459,962	1,736,867

Note. ASSIST2009[1]: The dataset is sourced from the Assessment's online education platform, extensively utilized for knowledge tracking. This educational system aids students in problem-solving and enhancing their learning capabilities. Data lacking concept records were excluded from this dataset.

Bridge Algebra2006[3]: The dataset, known as Bridge to Algebra 2006-2007, originates from the KDD Cup 2010 EDM challenge. It is derived from algebra courses offered on the Cognitive Algebra Tutor System.

In our experiments, we exclusively utilize sequences with a length exceeding 3, as shorter sequences lack meaningfulness. Each dataset is partitioned such that 80% of all sequences constitute the training set, while the remaining 20% form the test set.

Evaluation

In the KT task, we simultaneously use accuracy (ACC) and the area under the curve (AUC) (Nahler, 2009) to evaluate GTFN. This comprehensively assesses the model's overall ability to predict whether students can master specific knowledge points at a given time point. The combined use of these two metrics helps us gain deeper insights into the model's performance in the KT task, guiding better model improvement and optimization.

Baselines

DKT

DKT utilizes RNN to model students' knowledge states. However, it only constructs the graph structure using question nodes, without considering other effective features (Piech et al., 2015).

GKT

This is a KT model based on GNN, transforming knowledge states into a graph structure for predicting student performance (Nakagawa et al., 2019). At each time step, GKT updates the states by aggregating the states of neighboring nodes, while also updating the states of those neighboring nodes.

AKT

This KT network integrates attention mechanisms to simulate learners' past performance by constructing context-aware representations of questions and answers (Ghosh et al., 2020). However, its attention network has a shallow layer structure.

GIKT

This utilizes a GCN network to aggregate exercise-KC embedded features learned from high-order relationships (Yang Y. et al., 2020). It directly combines exercise embeddings with the corresponding answer embeddings as inputs to the KT model. However, this method does not take timing issues into consideration.

QDEKT

This model comprehensively considers both question difficulty and knowledge concept difficulty (Liu et al., 2022). It integrates both actual and predicted student responses to evaluate the student's knowledge acquisition ability and dynamically update the student's knowledge state. Additionally, the incorporation of additional features has been observed to greatly enhance predictive capabilities.

SHDKT

This introduces the concurrent skill graph and hierarchical skill representation module to explore the sequence and co-occurrence relationships between skills (Yang et al., 2022). Additionally, it introduces a problem representation module, learning explicit and implicit representations of problems through interaction with information related to the problems.

Experiment Setup

When implementing all comparative methods, we employ the TensorFlow framework. The embedding dimensions for both skills and questions are set to 100, and during the training process, all embedding matrices undergo random initialization and continuous updates. In the knowledge graph encoding layer, we set the maximum aggregation layer l of the GAT network to 3. Additionally, we utilize the Dropout with a retention probability of 0.8 (Zarafshani et al., 2016). Parameters are optimized using the Adam with a learning rate of 0.001, and the mini-batch size is set to 32 (Reyad et al., 2023). Other hyperparameters are chosen through multiple experiments, including the number of layers in GAT and the quantity of subspaces in multi-head attention.

Compared to the Baseline Models

To address the first question, a comparison was conducted between the proposed GTFN model and existing methods on different datasets, evaluating the performance of each model based on AUC and ACC values.

As shown in Table 4, on the three datasets, GIKT, which utilizes a graph structure, performs better than AKT, which employs an attention structure. The reason is that a graph structure often enables a deeper correlation of information features between nodes, whereas attention structures are constrained by the depth of the attention layers. The QDEKT model proves to be effective, as it introduces difficulty features of questions, alleviating the issue of treating different questions equally importantly for students. Additionally, it utilizes a sequence neural network to model student learning states, contributing to its superior performance. The SHDKT method extensively explores

Table 4. Results

Methods	ASSIST2009		ASSIST2012		Algebra2006	
	AUC	ACC	AUC	ACC	AUC	ACC
DKT	0.718	0.707	0.714	0.702	0.726	0.713
GKT	0.735	0.736	0.742	0.747	0.751	0.742
AKT	0.785	0.763	0.776	0.772	0.758	0.754
GIKT	0.781	0.767	0.773	0.771	0.764	0.761
QDEKT	0.786	0.772	0.778	0.769	0.782	0.776
SHDKT	0.792	0.784	0.781	0.778	0.793	0.780
GTFN	**0.801**	**0.795**	**0.796**	**0.791**	**0.807**	**0.789**

relationships between nodes in the skill graph, incorporating a hierarchical structure to capture high-order relationships between skills at different levels. Consequently, it achieves better results.

From this, constructing a graph structure and utilizing sequence neural network algorithms, as well as a multi-layered structure, can effectively improve predictive performance. Hence, the effectiveness of GTFN is more pronounced. The reason is that it not only constructs a graph structure for the relationship between questions and knowledge skills but also effectively employs a multi-layered GAT network. This network, when aggregating neighboring nodes, assigns different weights to different nodes. This means that the contributions of different neighbors can be treated differentially, unlike the GCN that treat neighboring nodes equally. The benefit is that it can better handle the heterogeneity between nodes, where different nodes may have different contributions to the task. Additionally, the adaptive selection of weights for neighboring nodes allows the model to be more flexible in learning tasks. Furthermore, GTFN introduces a GRU network, enabling dynamic modeling of student learning states over time, which helps to capture changes in students' knowledge at different time steps. Finally, using the last layer of the GAT network as the query matrix for attention allows the attention network to focus on the importance of nodes globally, so that it is not just limited to local neighbor information. Therefore, the performance of GTFN is superior to other baseline models.

Ablation Study

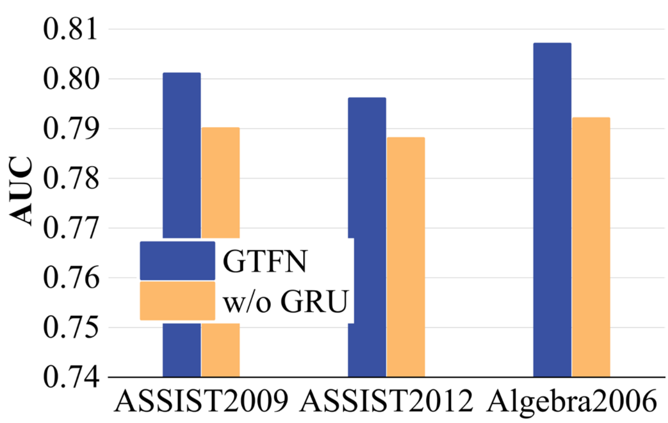
To address the second question, a series of ablation experiments are conducted to analyze the performance contributions of various parts of the model, as well as the relative importance of each component.

The Impact of the GRU Network Component

To explore the impact of the GRU method, *w/o GRU* denotes without using this method. This approach, as an RNN structure suitable for sequential data, effectively captures changes in student knowledge at different time steps. Through its gating mechanism, it aids in forgetting or retaining previous information, especially important learning contexts at historical time steps.

As depicted in Figure 2, removing this component significantly decreases the predictive performance of the model. This reduction may be attributed to the potential loss of adaptability to the temporal dynamics of students' learning trajectories, thereby diminishing the model's sensitivity to changes in students' knowledge states at different time points. The GRU, with its gating mechanism, efficiently captures and memorizes the historical information of students, and its removal makes it challenging to handle the impact of previous learning stages effectively.

Figure 2. The Impact of the GRU Network Component



Hence, preserving the GRU network is crucial for a comprehensive understanding of the temporal changes in students' learning states.

The Impact of Dynamic Attention Aggregation Network (DAAN)

To explore the impact of the DAAN on the experiment, *w/o DAAN* indicates the removal of the Dynamic Attention Aggregation Network. In fact, this network component consists of the last layer of the GAT network and the attention network, using the last layer as the query matrix for the attention network. As shown in Figure 3, GTFN with this component performs better because this design allows it to adjust its attention to different parts of the graph at each time step effectively, adapting to changes in students' learning states. This dynamic, graph-based attention mechanism helps the model better understand the complex relationships between problems and knowledge points, enhancing adaptability to the dynamic nature of students' learning trajectories.

However, if this design is removed, the model loses its ability to model students' learning states dynamically. Using the last layer's node representations as the query matrix enables the model to consider information from the entire graph comprehensively. Removing this design causes the model to rely more on local information, making it ineffective in capturing the correlations in different parts of the graph and, consequently, affecting the accurate prediction of students' learning states.

Hyperparameter Analysis

To address the third question, we explored the impact of certain hyperparameters of GTFN on the experiments.

The Impact of GAT Network Layer Depth

To explore the impact of GAT network layer depth, we set the number of layers to To [1, 2, 3, 4, 5]. The selection of GAT network layers involves understanding the complexity of student learning trajectories and the generalization ability of the model.

As shown in Figure 4, GTFN performs best when the depth is 3. First, the layer depth of the GAT network affects the modeling ability of complex relationships between problems and knowledge points. Increasing the number of layers helps to better understand the information in the graph structure but also introduces additional computational overhead. This not only captures students' answers to questions but also considers students' learning history and correlations between knowledge points.

Figure 3. The Impact of DAAN

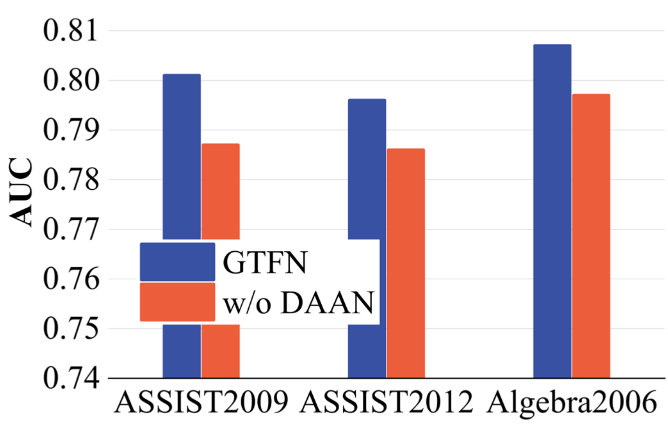
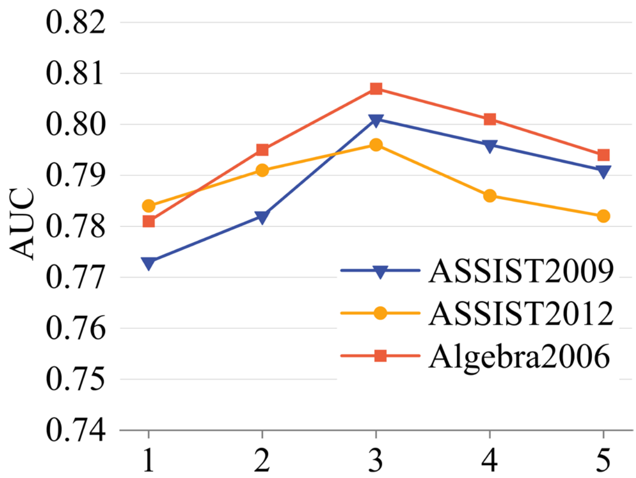


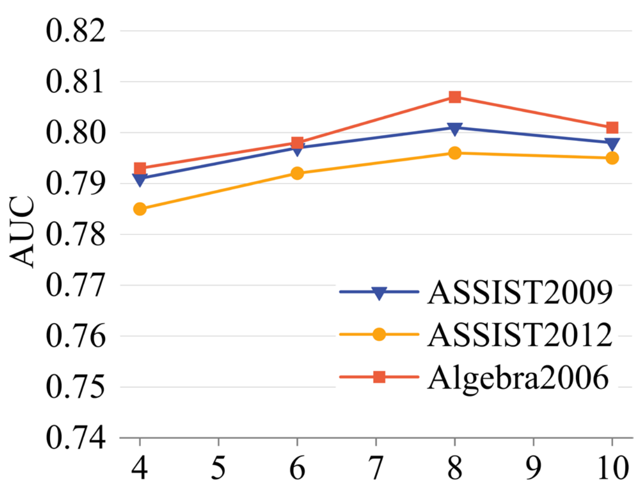
Figure 4. The Impact of GAT Network Layer Depth



The Influence of the Number of Subspaces

We set the number of subspaces to [4,6,8,10]. As shown in Figure 5, as the number of subspaces increases, the model effect is better, and the effect weakens when it exceeds 8. The number of subspaces in the multi-head attention network is a crucial parameter influencing the model's ability to capture complex relationships between questions and knowledge points. Each attention head can be considered a subspace with its own weight allocation mechanism, allowing it to focus on different aspects of the data simultaneously. A smaller number of subspaces may be limited when dealing with complex relationships, while a larger number can increase computational complexity and even lead to overfitting.

Figure 5. The Impact of the Number of Subspaces



DISCUSSION

Advantages

We proposed a KT model based on Graph Temporal Fusion Networks, effectively leveraging the GAT network to capture the intricate relationships between questions and knowledge points successfully. This enhancement allows the model to grasp the node correlations more efficiently in learning tasks, improving the understanding of questions and knowledge points. The introduction of the GRU network enables dynamic modeling of students' learning states over time, aiding in better capturing changes in students' knowledge at different time steps and enhancing adaptability to the dynamic nature of students' learning trajectories. Finally, using an attention network to aggregate node representations, we successfully integrated information from different levels.

Limitations

However, the GTFN model still has limitations. The use of multi-head attention networks in the GAT layer increases algorithmic complexity. In future research, we plan to resolve and improve this aspect. Additionally, there are other features that could be explored for KT tasks, such as textual information. In our upcoming work, we will explore these directions for further enhancement.

CONCLUSION

We proposed a GTFN approach, leveraging GAT networks to capture the intricate relationships between questions and knowledge points. We introduced GRU networks for dynamic temporal modeling of student learning states and employed an attention network to aggregate node representations, enhancing adaptability to student learning trajectories. However, the use of multi-head attention networks may result in increased algorithmic complexity. In the next phase of our work, we plan to address and improve this aspect while exploring other available features such as textual information.

CONFLICTS OF INTEREST

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

FUNDING STATEMENT

No funding was received for this work.

PROCESS DATES

Received: 1/9/2024, Revision: 3/18/2024, Accepted: 3/18/2024

CORRESPONDING AUTHOR

Correspondence should be addressed to Meng Huang; 13659211294@163.com

REFERENCES

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

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

Cui, C., Yao, Y., Zhang, C., Ma, H., Ma, Y., Ren, Z., Zhang, C., & Ko, J. (2024). DGEKT: A dual graph ensemble learning method for knowledge tracing. *ACM Transactions on Information Systems*, *42*(3), 1–24. doi:10.1145/3638350

Cui, P., Wang, X., Pei, J., & Zhu, W. (2017). A Survey on Network Embedding. *IEEE Transactions on Knowledge and Data Engineering*, *31*(5), 833–852. doi:10.1109/TKDE.2018.2849727

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

Han, X., Zhang, S., Zhou, J., Li, Z., & Wang, J. (2022). Deep knowledge tracing with GRU and learning state enhancement. In *Proceedings of the International Conference on Machine Learning for Cyber Security*. Springer.

Huang, W., Cai, S., Li, H., & Cai, Q. (2023). Structure graph refined information propagate network for aspect-based sentiment analysis. *International Journal of Data Warehousing and Mining*, *19*, 1–20. doi:10.4018/IJDWM.327363

Lasheng, Y. (2020). Research on deep knowledge tracking incorporating rich features and forgetting behaviors. [New Series]. *Journal of Harbin Institute of Technology*, •••, 1–6.

Li, L., Doroslovački, M., & Loew, M. H. (2020). Approximating the gradient of cross-entropy loss function. *IEEE Access : Practical Innovations, Open Solutions*, *8*, 111626–111635. doi:10.1109/ACCESS.2020.3001531

Liu, H., Peng, W., & Qi, C. (2022). Knowledge tracing with multi-feature fusion and question difficulty. In *Proceedings of the 2022 IEEE 21st International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS)*, 231-238. IEEE. doi:10.1109/IUCC-CIT-DSCI-SmartCNS57392.2022.00045

Minn, S., Yu, Y., Desmarais, M. C., Zhu, F., & Vie, J. (2018). Deep knowledge tracing and dynamic student classification for knowledge tracing. In *Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM)*, 1182-1187. IEEE. doi:10.1109/ICDM.2018.00156

Nahler, G. (2009). area under the curve (AUC).

Nakagawa, H., Iwasawa, Y., & Matsuo, Y. (2019). Graph-based knowledge tracing: Modeling student proficiency using graph neural network. In *Proceedings of the 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, 156-163. Association for Computing Machinery. doi:10.1145/3350546.3352513

Pandey, S., & Srivastava, J. (2020). RKT: Relation-aware self-attention for knowledge tracing. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. Association for Computing Machinery.

Pardos, Z. A., & Heffernan, N. T. (2010). Modeling individualization in a bayesian networks implementation of knowledge tracing. In *Proceedings of the International Conference on User Modeling, Adaptation, and Personalization*. Springer. doi:10.1007/978-3-642-13470-8_24

Park, W., & Kwon, H. (2023). Implementing artificial intelligence education for middle school technology education in Republic of Korea. *International Journal of Technology and Design Education, 34*, 1-09-135.

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

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

Su, Y., Cheng, Z., Luo, P., Wu, J., Zhang, L., Liu, Q., & Wang, S. (2021). Time-and-concept enhanced deep multidimensional item response theory for interpretable knowledge tracing. *Knowledge-Based Systems*, *218*, 106819. doi:10.1016/j.knosys.2021.106819

Tu, L., Zhu, X., & Ji, Y. (2023). Graph-based dynamic interactive knowledge tracing. In *Proceedings of the 2023 8th International Conference on Distance Education and Learning*, 49-56. Association for Computing Machinery. doi:10.1145/3606094.3606124

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. *stat, 1050*(20), 10-48550.

Yang, S., Zhu, M., Hou, J., & Lu, X. (2020). Deep knowledge tracing with convolutions. *ArXiv, abs/2008.01169*.

Yang, X., Ma, H., & Wang, M. (2023). Research on rumor detection based on a graph attention network with temporal features. *International Journal of Data Warehousing and Mining*, *19*(2), 1–17. doi:10.4018/IJDWM.319342

Yang, Y., Shen, J., Qu, Y., Liu, Y., Wang, K., Zhu, Y., . . . Yu, Y. (2021). GIKT: a graph-based interaction model for knowledge tracing. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I* (pp. 299-315). Springer International Publishing. doi:10.1007/978-3-030-67658-2_18

Yang, Z., Xu, S., Wang, C., & He, G. (2022). Skill-Oriented Hierarchical Structure for Deep Knowledge Tracing. In *Proceedings of the 2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI)*, 425-432. IEEE. doi:10.1109/ICTAI56018.2022.00069

Zarafshani, K., Sharafi, L., Binaian, A., Malekhosseini, A., & Norouzi, M. (2016). Determining factors that influence dropout rates in agricultural college at razi university. *Quarterly Journal of Research and Planning in Higher Education*, *1395*, 19–43.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., ... & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI open, 1*, 57-81.

ENDNOTES

1 https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data/skill-builder-data-2009-2010
2 https://sites.google.com/site/assistmentsdata/datasets/2012-13-school-data-with-affect
3 https://www.kdd.org/kdd-cup/

Huang Meng, Lecturer, Ph.D, graduated from the School of Humanities, Xi ‘an Jiao Tong University in 2015. Now she worked at the School of Marxism, Xi ‘an Technological University. Research interests include Cultural Philosophy, Philosophy of Information.

Wei Ting, Engineer, Master’s degree, graduated from Xi ‘an Technological University in 2004.Now he worked at the School of Computer Science and Technology, Xi ‘an Technological University. Research interests include Computer Application, Local Area Network Construction.