Frontiers of Digital Education
https://doi.org/10.1007/s44366-026-0097-8
Front. Digit. Educ. 2026, 3(3): 23
RESEARCH ARTICLE

CN 10-1952/G4
ISSN 2097-3918

Unveiling the Impact of Hierarchical Knowledge Dependencies on Knowledge Tracing: A Spatial Structure Perspective

Yuang WEI[a,b], Rui JIA[a] , Yingwen DING[a], Bo JIANG[a] (✉)

[a] Shanghai Institute of Artificial Intelligence for Education, East China Normal University, Shanghai 200062, China

[b] Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan 430079, China

✉ *Corresponding author. E-mail: bjiang@deit.ecnu.edu.cn (Bo JIANG).*

Front. Digit. Educ., **Just Accepted Manuscript** • DOI: 10.1007/s44366-026-0097-8

http://journal.hep.com.cn on June 5, 2026

Unveiling the Impact of Hierarchical Knowledge Dependencies on Knowledge Tracing: A Spatial Structure Perspective

Abstract

Knowledge tracing predicts learners' evolving knowledge state by tracking their performance over time. While temporal dynamics have been the focus of most KT studies, the spatial structure among knowledge components (KCs) remains underexplored despite its rich latent information. Prior work suggests that inter-KC relationships can enhance knowledge tracing performance, yet the impact of hierarchical spatial structures is still unclear. This study investigates how multi-level spatial relationships among KCs affect the performance of knowledge tracing models. Using causal structure learning, we infer causal links among KCs and incorporate the resulting spatial structures into both deep learning and traditional machine learning knowledge tracing models. Experimental results show that incorporating second-order spatial structures yields consistent performance gains. These findings underscore the value of spatial structural information in knowledge tracing. Furthermore, interpretable feature analyses illustrate how spatial features shape diagnostic predictions, providing insight into factors underlying students' learning challenges. This spatial perspective not only improves knowledge tracing model performance but also has the potential to inform more targeted and effective instructional strategies.

Keywords: Knowledge Tracing, Spatial Structure, Explainable AI, SHAP

1 Introduction

Mastery of knowledge components (KCs) provides an explicit representation of a learner's cognitive state, and accurately identifying this state is fundamental to achieving personalized learning (Ritter, Yudelson, Fancsali, & Berman, 2016). A critical aspect of this identification is understanding the features and factors that influence learners' mastery of KCs, as these are key to precisely measuring cognitive states (M. Zhang et al., 2023). Existing work in cognitive diagnosis, particularly in knowledge tracing (KT), has mainly focused on analyzing temporal sequences of learning behaviors to uncover important elements of personalized learning, such as changes in cognitive states and memory decay (Nagatani et al., 2019; Wei & Jiang, 2023). In addition, some studies have modeled relational structures among KCs to extract

behavioral associations (Nakagawa, Iwasawa, & Matsuo, 2019; Wei, Zhou, Jiang, & Jiang, 2024; Yang et al., 2021). Despite these advances, the multi-level, hierarchical organization of KCs remains underexplored. In this work, we refer to this organization as the *spatial structure* of KCs. We use "spatial" in a graph-theoretic rather than physical sense: KCs and their prerequisite or causal relations form a directed graph, where vertices represent KCs and edges encode directed dependencies (Nakagawa et al., 2019; Tong et al., 2020). For a target KC, its spatial structure is defined as the multi-hop neighborhood around it in this graph (e.g., its parents, children, and higher-order ancestors and descendants), which jointly captures hierarchical organization, prerequisite links, and causal pathways among KCs (Abdelrahman, Wang, & Nunes, 2023; Jiang, Wei, Zhang, & Zhang, 2024). However, many cognitive diagnostic models still treat KCs as independent entities. Even when structural relationships among KCs are considered, potential hierarchical dependencies are often ignored (Abdelrahman et al., 2023). Such simplifications can limit our understanding of interactions among KCs and lead to suboptimal accuracy when assessing a learner's mastery (Nakagawa et al., 2019). In particular, it often remains unclear which KCs are most critical for mastering a specific learning objective, or how many instructional interventions are required to achieve substantial learning gains.

Beyond predictive performance, interpretability is also essential. Whether the identified features genuinely affect mastery of a target KC, which features exert the strongest influence on the assessment of that target, and why they have such effects are key questions for gaining the trust of teachers and students in diagnostic results. These questions remain insufficiently answered in current KT research.

To address these gaps, this study explicitly introduces hierarchical spatial structures among KCs into KT. By defining and incorporating first-order, second-order, and third-order nodes, we capture multi-level dependencies among KCs and integrate the resulting spatial structures into cognitive diagnostic models. Experimental results show that leveraging these structures can improve diagnostic performance. In particular, second-order spatial features yield the most favorable overall results, with the largest improvements observed in the interpretable feature-based model and more modest but generally positive gains in deep KT models. These findings highlight the important role of hierarchical dependencies among KCs in learners' mastery of target concepts. They further suggest that effective instructional interventions may benefit from identifying second-order parent nodes of the current KC to address learning difficulties at their source, whereas focusing on more distant nodes may offer limited benefits and intervening only in immediately adjacent nodes may not fully resolve the problem.

To meet the need for interpretability, we further employ SHAP-based analysis methods (Lundberg & Lee, 2017) to examine how features of KCs within the learned hierarchy influence mastery of target concepts. Through this visualization-based interpretive approach, we can observe the contribution of different features and identify which KCs within the spatial structure are most influential for a given target. This interpretability analysis not only deepens our understanding of hierarchical knowledge

structures, but also provides guidance for designing more transparent educational models. By highlighting key drivers in the learning process, our approach offers insights that can enhance the practical utility and perceived credibility of KT models.

Specifically, our study makes three main contributions:

- We systematically investigate how hierarchical knowledge structures affect learning outcomes within a cognitive diagnosis framework.
- We propose a transparent and interpretable KT framework that can be instantiated with a variety of KT models.
- We show that incorporating spatial structural information can improve both the performance and interpretability of deep KT models.

2 Related Work

2.1 Methods for Improving Knowledge Tracing Models

The main task of KT is to predict learners' mastery levels of various KCs based on the sequence of their responses (correct or incorrect) (Abdelrahman et al., 2023). Essentially, it involves dynamic tracing of learners' knowledge states, and recent studies have further extended this process by considering affective dynamics during learning (Sun, Zhang, et al., 2025). KT models are typically divided into two categories: traditional and deep knowledge tracing models.

Early KT studies mainly relied on probabilistic models such as BKT (Corbett & Anderson, 1994) and its personalized variants (Yudelson, Koedinger, & Gordon, 2013), as well as IRT- and factor-analysis-based models, including HIRT (Wilson, Karklin, Han, & Ekanadham, 2016), AFM (Cen, Koedinger, & Junker, 2008), and PFA (Pavlik, Cen, & Koedinger, 2009). Recent studies have increasingly adopted deep-learning architectures, including DKT (Piech et al., 2015), DKVMN (J. Zhang, Shi, King, & Yeung, 2017), SAKT (Pandey & Karypis, 2019), and GKT (Nakagawa et al., 2019), to model learners' temporal knowledge states and KC relationships. Although these deep KT models improve predictive capacity, they often provide limited structural transparency and explainability, especially in revealing how hierarchical dependencies among KCs affect learners' mastery.

Building on the evolution of KT models, researchers have explored three main directions to improve KT performance: knowledge graph-based modeling, knowledge structure modeling, and feature selection (Sun, Liu, et al., 2025). Knowledge graph-based methods introduce graph structures into KT, where GNNs (Scarselli, Gori, Tsoi, Hagenbuchner, & Monfardini, 2008) and models such as GIKT (Yang et al., 2021) are used to capture dependencies between KCs and interactions between questions and knowledge elements, though they often rely on manually constructed graphs and complex neural operations. Knowledge structure modeling further captures richer KC or exercise relationships, as in SKT (Tong et al., 2020) and EKT (Liu et al., 2019), but their information propagation and memory mechanisms still make the reasoning process difficult to interpret. Feature selection methods, such as KTM (Vie & Kashima, 2019) and KTM-DLF (Gan, Sun, Peng, & Sun, 2020), model interactions among learners, questions, and skills with relatively better efficiency and interpretability, yet they

may be less expressive in capturing complex KC relationships. Overall, the diverse methods combining knowledge graphs, knowledge structures, and feature selection provide new perspectives and tools for KT. Despite the promising potential of these models in enhancing performance and personalized learning path recommendations, they generally exhibit insufficient explainability, which limits their transparency and trustworthiness in practical applications.

2.2 Explainable Knowledge Tracing Models

In the field of KT, existing methods have made significant progress in terms of explainability; however, these efforts are often limited to the specific model structures, lacking a universally applicable framework for explanation. For instance, the Bayesian Knowledge Tracing (BKT) model is known for its simple structure, allowing parameters such as learning rate and forgetting rate to explain changes in learners' knowledge states. Nevertheless, BKT overlooks long-term dependencies and forgetting phenomena, which limits its ability to capture complex learning behaviors. Similarly, Learning Factor Analysis (LFA) (Cen, Koedinger, & Junker, 2006) is easy to explain but assumes a uniform learning rate across all learners and KCs, thus lacking adaptability to individual learning patterns.

More complex models like Dynamic Key-Value Memory Networks (DKVMN) and Convolutional Knowledge Tracing (CKT) (Shen et al., 2020) can handle forgetting and capture interactions between learners and KCs, but their complexity and long training times make them difficult to deploy in practical settings. Additionally, CKT provides implicit personalized measures but lacks explicit explainability, making it challenging for users to understand the rationale behind model predictions. Similarly, Attention-Based KT uses attention mechanisms to explain the influence of learning interactions but suffers from complexity that limits the intuitive understanding of attention weight calculations.

In the realm of machine learning explainability research, SHAP (Lundberg & Lee, 2017) (Shapley Additive Explanations) offers an innovative game-theoretic approach designed to explain model outputs. Developed by Lundberg and Lee, SHAP provides a unified framework for explaining different types of machine learning algorithms. By calculating the marginal contribution of each feature to the model's prediction, SHAP ensures fairness and consistency in its explanations. With properties such as consistency, missingness handling, and local accuracy, SHAP's model-agnostic nature makes it applicable across various machine learning techniques. This makes SHAP particularly advantageous in scenarios where transparency and explainability are crucial, especially in education.

In educational research, Nnadi et al. (Nnadi, Watanobe, Rahman, & John-Otumu, 2024) applied SHAP to predict learners' adaptability. Their study revealed that factors such as "class duration" and "financial condition" significantly influence learners' adaptability levels. These findings provide valuable insights for educational institutions, helping them design targeted interventions to enhance learners' adaptability. This SHAP-based analysis improves data transparency in education and offers clearer guidance for educational decision-making.

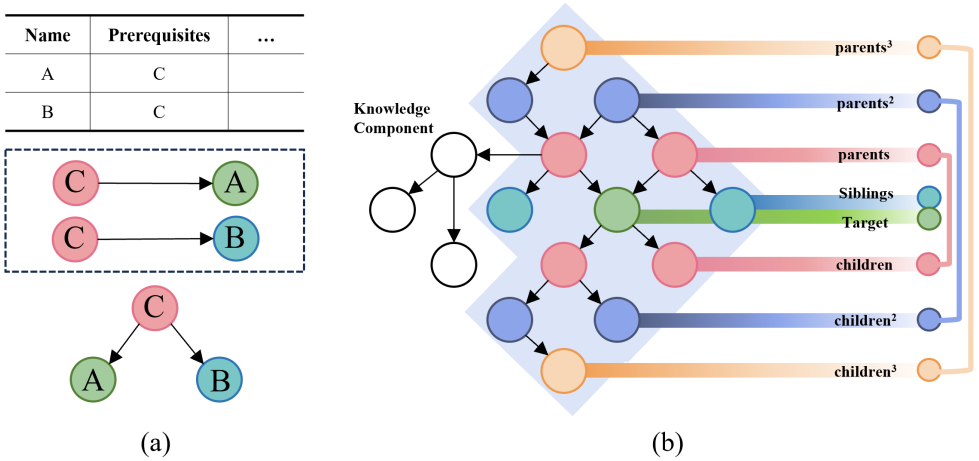

Fig. 1: The multi-level spatial structure of knowledge components

In summary, although existing KT methods have achieved certain success in their own model-specific explainability, the lack of a unified framework remains a key challenge. SHAP presents a promising solution that could further advance the field of KT.

3 Method

3.1 Knowledge Component Network Structure

To investigate the impact of hierarchical dependencies among KCs on KT methods, we first need a clear description of how KCs are connected. In this paper, we model the KC network as a directed graph: each KC is a node, and each prerequisite or causal relation is a directed edge between two nodes. In this graph, the distance between two KCs is defined as the number of edges on the shortest path connecting them.

Let us begin by defining the structural relationships within the network. As shown in Figure 1(a), if node A has a preceding node C, then C and A form a parent-child relationship. When C serves as a common parent to both A and B, A and B are sibling nodes.

In addition, considering the multi-level influence between KCs, we categorize the structural space into first-order, second-order, and third-order neighborhoods around the target KC. In this study, the notion of order mainly reflects the depth of parent-child propagation in the KC dependency graph. Sibling nodes are not necessarily one edge away from the target node in a strict graph-theoretic sense, but they are treated as immediate pedagogically related nodes because they share the same parent with the target KC. Following work on the zone of proximal development (ZPD) in educational psychology and its computational operationalizations (Chaiklin, 2003; Chounta, Albacete, Jordan, Katz, & McLaren, 2017), we assume that instructional effects are mainly driven by knowledge that is conceptually close to the learner's

current competence, while very distant concepts contribute much less. In our setting, graph distance in the KC dependency network serves as a proxy for such conceptual distance. Therefore, we restrict the spatial neighborhood of a target KC to nodes within three degrees of separation. As shown in Figure 1(b), this is defined as follows:

We focus on the parent nodes, child nodes, and sibling nodes of the target KC. The set of parent, child, and sibling nodes is termed the First-order Immediate nodes. Parent and child nodes are directly connected to the target KC, while sibling nodes are included because they share an immediate parent with the target KC:

$$N_{\text{First-Im}} = \{N_{\text{parents}}, N_{\text{children}}, N_{\text{siblings}}\}. \tag{1}$$

Considering the parent nodes (*parents*), second-level parent nodes (parents^2), child nodes (*children*), second-level child nodes (children^2), and sibling nodes (*siblings*) of the target KC, we define the collection of these five types of nodes as Second-order Immediate nodes, denoted as:

$$N_{\text{Second-Im}} \triangleq \{\, N_{\text{parents}},\, N_{\text{parents}^2},\, N_{\text{children}},\, N_{\text{children}^2},\, N_{\text{siblings}}\}. \tag{2}$$

Considering the parent nodes (*parents*), second-level parent nodes (parents^2), third-level parent nodes (parents^3), child nodes (*children*), second-level child nodes (children^2), third-level child nodes (children^3), and sibling nodes (*siblings*) of the target KC, we define the collection of these seven types of nodes as Third-order Immediate nodes, denoted as:

$$N_{\text{Third-Im}} = \left\{ \begin{array}{l} N_{\text{parents}},\, N_{\text{parents}^2},\, N_{\text{parents}^3}, \\ N_{\text{children}},\, N_{\text{children}^2},\, N_{\text{children}^3},\, N_{\text{siblings}} \end{array} \right\}. \tag{3}$$

After defining the structure of the KC network, the next step is to learn the structure of the causal network among the KCs. We first consider a set of KCs $\mathcal{K} = \{K_1, K_2, \ldots, K_n\}$, where each KC K_i corresponds to a random variable. The goal is to learn a directed acyclic graph (DAG) $G = (V, E)$ from the observed data $\mathbf{D}$, where V represents the set of nodes corresponding to the KCs $\mathcal{K}$, and E represents the set of directed edges that denote the causal relationships between the KCs.

The initial graph G_0 could either be a prior network structure inherent in the data or a structure provided by experts. In this graph, each node represents a KC. Next, we define a score function $S(G, \mathbf{D})$, such as the Bayesian Information Criterion (BIC) used in this study, to evaluate the quality of the graph G. The BIC score is calculated as follows:

$$\text{BIC}(G, \mathbf{D}) = \log P(\mathbf{D} \mid G) - \frac{\log N}{2} \cdot \text{Dim}(G) \tag{4}$$

In this context, $\log P(\mathbf{D} \mid G)$ represents the log-likelihood of the data $\mathbf{D}$ given the graph G, N is the sample size, and $\text{Dim}(G)$ refers to the number of parameters in graph G.

To compute the graph, we start by setting the current graph to G_0. Next, for each possible edge $(K_i \rightarrow K_j)$ in the current graph, we attempt to add it and compute the score $S(G', \mathbf{D})$ for the new graph G' formed after adding the edge. If the score

$S(G', \mathbf{D})$ exceeds $S(G, \mathbf{D})$, we accept the new edge and update the graph to G'. This process continues until no further edge additions improve the score.

Afterward, a local search is conducted among the equivalent classes of the current graph (i.e., all graphs statistically equivalent to the current one), and the graph with the highest score is selected to update the current graph.

Next, for each possible edge $(K_i \to K_j)$ in the current graph G, we attempt to delete the edge and compute the score $S(G', \mathbf{D})$ for the new graph G' after the deletion. If $S(G', \mathbf{D})$ exceeds $S(G, \mathbf{D})$, the deletion is accepted, and the graph is updated to G'. This process continues until no further edge deletions improve the score.

Similarly, a local search is performed within the equivalent class of the current graph, and the graph with the highest score is chosen. The algorithm terminates when neither the forward (addition) nor backward (deletion) steps can further improve the score. The final graph G is the learned causal network structure of the KCs. The detailed algorithm is shown in Algorithm 1.

We employ the Algorithm 1 to infer a directed graph over KCs from observational student response data. Following standard causal discovery practice, fGES assumes (i) causal sufficiency (no unmeasured common causes among the observed variables), (ii) acyclicity of the underlying causal graph, (iii) the causal Markov and faithfulness conditions, and (iv) correct model specification. These assumptions provide a useful starting point for recovering a plausible dependency structure, but they are unlikely to hold perfectly in real educational environments, where latent factors such as general ability, motivation, teacher effects, or curriculum design may influence multiple KCs simultaneously.

Consequently, in this paper, the graph learned by Algorithm 1 should be interpreted as a data-driven approximation to the underlying dependency structure, or as a set of candidate causal hypotheses, rather than as a definitive map of instructional prerequisites. Some edges may arise from unobserved confounders or shared instructional contexts rather than direct prerequisite relations, and some true dependencies may be missed. In this paper, we therefore use the fGES-based structure primarily as a causally informed representation to construct spatial features for KT, and we are cautious in drawing strong causal conclusions from individual edges.

3.2 Knowledge Tracing Model

We consider two feature-encoding schemes regarding the information of related KCs. In Assumption 1, each correct response to a related KC receives a positive score of (+1), while each incorrect response receives a score of (0). This scheme can be interpreted as a standard learning outcome that only accumulates successful responses. In Assumption 2, each correct response to a related KC receives a positive score of (+1), while each incorrect response receives a penalty score of (-1). This scheme can be seen as a learning outcome with penalty, as it considers both successful and unsuccessful responses. For each type of related KC, the scores are aggregated based on the learner's historical response records. We take the first-order features as an example, as shown in Table 1.

Subsequently, using the causal relationship structure between KCs obtained in Sections 3.1, as well as the multi-level KC information affecting the target nodes, we

Algorithm 1 High-level procedure of fGES-based KC dependency structure learning

Require: Data $\mathbf{D}$, Initial Graph G_0, Scoring Function $S(G, \mathbf{D})$
Ensure: Learned Graph G

$G \leftarrow G_0$
Forward Phase:
while true **do**
    $G_{\text{best}} \leftarrow G$
    for each edge $(K_i \rightarrow K_j)$ not in G **do**
        $G' \leftarrow G \cup \{K_i \rightarrow K_j\}$
        if $S(G', \mathbf{D}) > S(G_{\text{best}}, \mathbf{D})$ **then**
            $G_{\text{best}} \leftarrow G'$
        end if
    end for
    if $G_{\text{best}} = G$ **then**
        break
    else
        $G \leftarrow G_{\text{best}}$
    end if
end while
Backward Phase:
while true **do**
    $G_{\text{best}} \leftarrow G$
    for each edge $(K_i \rightarrow K_j)$ in G **do**
        $G' \leftarrow G \setminus \{K_i \rightarrow K_j\}$
        if $S(G', \mathbf{D}) > S(G_{\text{best}}, \mathbf{D})$ **then**
            $G_{\text{best}} \leftarrow G'$
        end if
    end for
    if $G_{\text{best}} = G$ **then**
        break
    else
        $G \leftarrow G_{\text{best}}$
    end if
end while
return G

incorporate this additional information into the deep knowledge tracing model. This step examines whether the additional information enhances the predictive performance of the deep knowledge tracing model. The detailed algorithm process is shown in Algorithm 2.

Combining deep learning models can achieve better accuracy; however, to explore whether this method can also be applied to interpretable models, we construct an interpretable KT model using white-box machine learning models. As an example, we use decision trees, and the algorithm process is shown in Algorithm 3.

Table 1: Features in the two types of assumptions.

Method	Feature	Description
I	*children_corrects*	The correct score of the child node
	parents_corrects	The correct score of the parent node
	siblings_corrects	The correct score of the sibling node
II	*children_sum*	The score of the child nodes
	parents_sum	The score of the parent nodes
	siblings_sum	The score of the sibling nodes
	correct_sum	The total score for correct answers
	incorrect_sum	The total score for incorrect answers

Algorithm 2 Integrating Causal Structures and Multi-Level Knowledge into Deep Knowledge Tracing Model

Require: Data $\mathbf{D}$, Causal Graph G, Target Node K_t, Deep Knowledge Tracing Model $\mathcal{M}$
Ensure: Updated Model $\mathcal{M}'$
Step 1: Extract Knowledge
Identify ancestors of K_t in G and their levels
Step 2: Encode Knowledge
Construct feature vectors $\mathbf{f}_t$ incorporating causal and multi-level info
$\mathbf{f}_t \leftarrow \text{Encode}(G, \text{Multi-Level})$
Step 3: Integrate with Model
Update input layer of $\mathcal{M}$ to include $\mathbf{f}_t$
Augment dataset: $\mathbf{D}' \leftarrow (\mathbf{D}, \mathbf{f}_t)$
Train $\mathcal{M}$ on $\mathbf{D}'$
Step 4: Optimize Model
Optimize $\mathcal{M}$ with backpropagation
Evaluate and fine-tune $\mathcal{M}'$
return Updated model $\mathcal{M}'$

4 Experiments

4.1 Dataset

This study uses the Junyi2015 dataset for experiments. Compared to other publicly available datasets such as ASSISTMENT and EdNet, the Junyi2015 dataset is similar in terms of the elements involved in the learning process and is considered a high-quality dataset. However, what sets the Junyi2015 dataset apart is that it contains prior knowledge information, including prerequisite relationships between KCs, which is one of the essential conditions for the experiments in this study. Therefore, the Junyi2015 dataset was chosen as the experimental dataset.

The Junyi2015 dataset includes the Junyi_ProblemLog_original, which is a practice log table from Junyi Academy. It contains a total of 25,925,992 practice records from 247,606 learners on 722 questions. Additionally, the Junyi_Exercise_Table records detailed information about 819 questions. Attributes related to the practice problems include name, topic, and area, where these attributes are organized hierarchically from smallest to largest. Each topic contains several names, and each area contains multiple topics. Topics are represented as larger nodes in the knowledge map. The specific relationships are shown in Table 2.

Algorithm 3 Constructing a Decision Tree Model with Causal and Multi-Level Knowledge

Require: Dataset $\mathbf{D}$ with features $\mathbf{X}$, Target variable Y, Causal Graph G, Target Node K_t
Ensure: Decision Tree Model $\mathcal{T}'$
Step 1: Extract Knowledge
Identify parents and higher-order ancestors of K_t in G
Group them into sets $\{A_1, A_2, \ldots, A_m\}$
Step 2: Encode Information
Create features $\mathbf{f}_t$ from ancestor data
Augment dataset: $\mathbf{D}' \leftarrow (\mathbf{X}, \mathbf{f}_t, Y)$
Step 3: Prepare Features
Normalize or standardize features $\mathbf{f}_t$
Optionally select relevant features
Step 4: Train Model
Initialize and train decision tree $\mathcal{T}$ using $\mathbf{D}'$
Step 5: Evaluate
Assess model performance
Optimize hyperparameters
Step 6: Deploy
Analyze and deploy trained model $\mathcal{T}'$
return Decision Tree Model $\mathcal{T}'$

Table 2: The feature structure of the Junyi dataset

Area	Topic	Exercise Name	
area X	topic A	name 1	name 2
	topic B	name 3	name 4
area Y	topic C	name 5	name 6
		name 7	

This study requires the data to meet two basic conditions: (1) the selected KCs must have corresponding answer records; and (2) the answer records must be sufficiently numerous. After performing statistical filtering on the dataset, we selected the data from the "geometry" area, obtained from the Junyi_Exercise_table table, which includes 161 KCs. By cross-referencing the exercise names with the Junyi_ProblemLog_original table, we found that 15 of these KCs were not present in the Junyi_ProblemLog_original table. These KCs were removed, leaving a final selection of 2,922,789 answer records.

4.2 Experimental Setup

For all deep knowledge tracing models (DKT, DKVMN, SAKT, KQN, GKT, and GIKT), we followed commonly adopted configurations in prior work to ensure fair comparisons. Specifically, DKT used an embedding dimension of 256 and a hidden state dimension of 500; DKVMN used a state embedding dimension of 256 with a memory size of 20; SAKT used an embedding dimension of 100 with 5 attention heads and a dropout rate of 0.2; KQN adopted value and state embedding dimensions of 256 with a hidden size of 256; GKT used a hidden state dimension of 256 with 8 attention

heads and the PAM propagation method; and GIKT used a 256-dimensional hidden layer with an embedding size of 128 and a maximum sequence length of 30. All models were trained using the Adam optimizer with an initial learning rate of 0.0001 and a batch size of 128. Early stopping with a patience of 5 epochs was applied based on validation performance to prevent overfitting. Each experiment was repeated three times, and the average results were reported. SpaKT is an interpretable, feature-based KT model implemented with a decision tree, using aggregated hierarchical spatial KC features as input. It does not employ neural embeddings or sequence modeling, and standard decision tree hyperparameters were selected via validation.

All experiments were conducted on a single NVIDIA Quadro P2200 with 5GB memory. Training non-graph-based KT models (DKT, DKVMN, SAKT, and KQN) required approximately 20—40 minutes, while graph-based models (GKT and GIKT) were more computationally intensive and required approximately 2—3 hours to train. SpaKT incurred substantially lower computational cost than deep KT models, as it does not involve iterative backpropagation. In our experiments, SpaKT training typically completed within several minutes (approximately 2—5 minutes), with the majority of the runtime dominated by spatial feature construction rather than model fitting. Overall, the proposed approach achieves consistent performance improvements with acceptable computational cost.

4.3 Evaluation Metrics

This study constructs models under different KC scenarios and compares them with baseline models to evaluate their effectiveness. Three commonly used metrics are adopted: Accuracy, AUC (Area Under the Curve), and RMSE (Root Mean Square Error).

Accuracy measures the proportion of correctly predicted samples among all samples. It reflects the overall correctness of the prediction results and is defined as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}, \tag{5}$$

where TP and TN denote correctly predicted positive and negative samples, respectively, while FP and FN denote incorrectly predicted positive and negative samples. A higher Accuracy value indicates better prediction performance.

AUC represents the area under the ROC curve and is used to evaluate the overall discriminative ability of a classifier. Its value generally ranges from 0.5 to 1 for an effective classifier, with a larger AUC indicating better classification performance.

RMSE measures the deviation between predicted and actual values and is sensitive to large prediction errors. A smaller RMSE indicates more accurate predictions. It is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}. \tag{6}$$

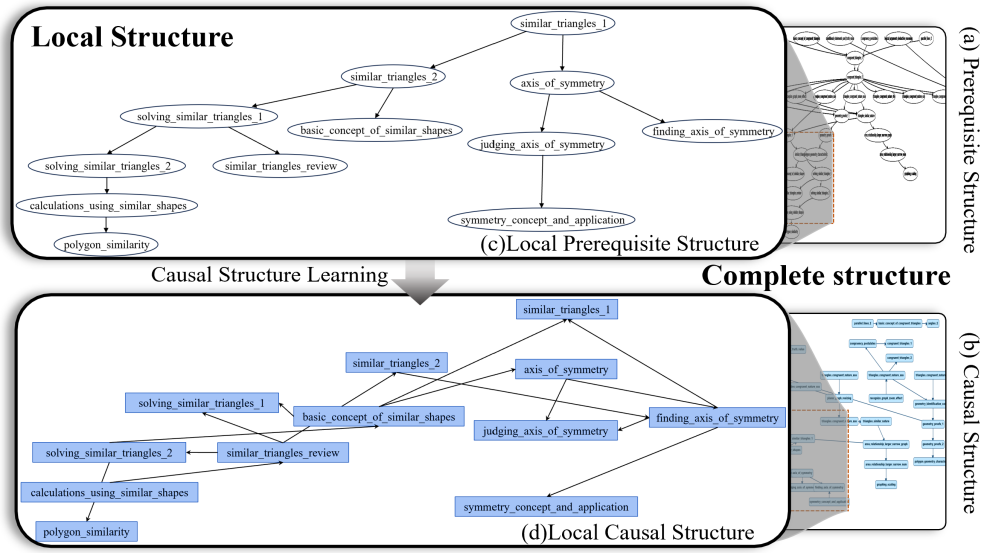

Fig. 2: The causal graph structure of knowledge components. (a) The original prerequisite relationship diagram of the knowledge components; (b) The causal structure diagram learned from data, using (a) as the initial graph; (c) (d) Sections of (a) and (b) are extracted for clearer viewing.

4.4 The Impact of Spatial Structure on Knowledge Tracing Performance

To validate the impact of multi-level KC information on the performance of KT, we conducted multiple experiments on various subgraphs within the geometry domain to explore which spatial structure can best improve KT performance. As an example, we use congruent_triangles_1 as the central KC to demonstrate how subgraphs are selected. The causal relationships between the KCs in the subgraph are obtained through the method outlined in Section 3.1, and the resulting causal structure is used for KT experiments.

Based on Figure 2, we present the knowledge space structure diagram centered on the KC congruent_triangles_1 (Figure 2(a)) as well as its corresponding causal structure diagram (Figure 2(b)). We use the local subgraph from these diagrams to explain the network relationships between the KCs in detail. The extracted local prerequisite knowledge graph, shown in Figure 2(c), contains 12 KCs and 11 prerequisite relationships. The corresponding causal structure graph, shown in Figure 2(d), contains 15 causal relationships. By comparing the network relationships of the local prerequisite graph with the causal relationships provided in Figure 2(d), we summarize the comparison between causal relationships and network relationships in Table 3. The first column represents the node at the tail of the arrow (Cause), the second column represents the node at the head of the arrow (Effect), and the third column (Relation)

Table 3: Comparison table of causal relationships and original prerequisite relationships

Cause	Effect	Relation
finding_axis_of_symmetry	similar_triangles_1	Second-order child → First-order parent
finding_axis_of_symmetry	judging_axis_of_symmetry	Sibling → Sibling, First-order parent
finding_axis_of_symmetry	symmetry_concept_and_application	Non-direct relationship → First-order parent
similar_triangles_review	solving_similar_triangles_1	First-order child → First-order parent
axis_of_symmetry	judging_axis_of_symmetry	First-order parent → First-order parent
axis_of_symmetry	finding_axis_of_symmetry	First-order parent → First-order parent
basic_concept_of_similar_shapes	solving_similar_triangles_1	Sibling → First-order parent
basic_concept_of_similar_shapes	similar_triangles_1	Second-order child → First-order parent
basic_concept_of_similar_shapes	axis_of_symmetry	Non-direct relationship → First-order parent
similar_triangles_review	similar_triangles_2	Second-order child → First-order parent
similar_triangles_review	solving_similar_triangles_2	Sibling → First-order parent
solving_similar_triangles_2	polygon_similarity	Second-order parent → First-order child
solving_similar_triangles_2	basic_concept_of_similar_shapes	Non-direct relationship → First-order parent
similar_triangles_2	finding_axis_of_symmetry	Non-direct relationship
calculations_using_similar_shapes	similar_triangles_review	Non-direct relationship → First-order parent

Table 4: Original Model Performance

Model	Accuracy(%)	AUC(%)	RMSE
DKT	81.63	72.13	0.1836
DKVMN	82.23	71.23	0.1776
SAKT	82.55	72.67	0.1744
KQN	81.99	67.93	0.1800
GKT	82.04	68.46	0.1796
GIKT	79.74	67.85	0.2051
SpaKT	64.32	71.32	0.5345

represents the changes in the node relationships from the prerequisite relationship diagram to the causal diagram.

Based on the causal structure learning procedure and the multi-level spatial feature construction described in Sections 3.1 and 3.2, we incorporate the resulting KC information into both deep learning and machine learning KT models. The corresponding performance comparisons are reported in Table 4 and Table 5.

In Table 4, we present the performance of models using raw information (such as question IDs, answer sequences, etc.), while in Table 5, we show the performance of models incorporating spatial feature information. A comparison reveals that the AUC of the DKT, DKVMN, and SAKT models increased by 2.1%, 1.4%, and 1.7%, respectively, after introducing spatial structure information. Most models achieve better AUC after incorporating second-order spatial information, although the improvements are not uniform across all metrics. Additionally, for SpaKT, compared to Method I in Table 1, which does not include penalty features, Method II in Table 1, which incorporates penalty features, achieved a 0.5% increase in AUC.

Based on these results, we can conclude: (1) Spatial information can enhance the model's performance to some extent; (2) Using second-order spatial information results in the best model performance; (3) Feature processing with penalties leads to better model performance.

Table 5: Performance After Incorporating Spatial Information

Model		Accuracy(%)			AUC(%)			RMSE		
	SpaS	n=1	n=2	n=3	n=1	n=2	n=3	n=1	n=2	n=3
DKT	cor	81.77	**81.83**	81.42	73.41	**73.70**	73.62	0.1822	**0.1816**	0.1857
	sum	81.38	**81.85**	81.54	72.04	**72.21**	71.98	0.1861	**0.1814**	0.1845
DKVMN	cor	82.14	**82.15**	82.15	71.44	**71.29**	71.26	0.1785	**0.1784**	0.1784
	sum	81.37	**81.66**	81.63	70.70	**72.26**	71.84	0.1862	**0.1833**	0.1836
SAKT	cor	82.43	**82.56**	82.49	72.77	**72.95**	72.62	0.1756	**0.1743**	0.1750
	sum	82.34	**82.44**	82.49	73.89	**73.96**	73.89	0.1765	**0.1755**	0.1750
KQN	cor	**83.06**	82.08	82.15	**67.98**	68.25	67.40	0.1793	0.1792	**0.1785**
	sum	81.98	82.01	**82.11**	67.27	**68.66**	67.47	0.1801	0.1799	**0.1789**
GKT	cor	82.08	**82.09**	82.01	68.58	**70.19**	69.96	**0.1791**	**0.1791**	0.1798
	sum	80.99	81.52	**82.06**	67.82	**72.39**	70.89	0.1901	0.1848	**0.1794**
GIKT	cor	79.83	**79.92**	80.02	68.25	**68.96**	67.14	0.2016	0.2008	**0.1998**
	sum	79.62	**80.13**	78.95	67.67	**69.22**	66.71	0.2040	**0.1987**	0.2104
SpaKT	cor	66.11	**72.42**	68.04	68.72	**74.81**	72.05	0.5822	**0.5391**	0.5901
	sum	71.70	**73.97**	70.01	73.31	**75.19**	73.51	0.5321	**0.5110**	0.5804

Note. SpaS indicates which spatial information was utilized. For example, when $n = 2$, it includes the parent-child nodes, second-order parent-child nodes, and sibling nodes. The term *cor* refers to using the features derived from Method I in Table 1, while *sum* refers to using the features derived from Method II in Table 1.

Firstly, the improvement brought by spatial information has been validated in both deep learning and machine learning methods. Secondly, the reason why second-order spatial information leads to the best model performance is likely due to the fact that second-order nodes contain critical information that is not present in first-order nodes, whereas third-order nodes introduce some redundant information, which negatively impacts the prediction of target nodes. Finally, in the process of identifying students' responses to questions, incorrect answers also contribute to prediction accuracy. Therefore, incorporating both correct and incorrect response statistics performs better than using only correct information alone.

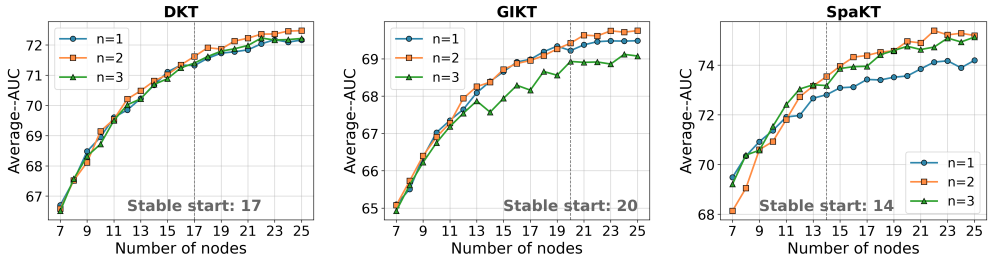

Fig. 3: Average AUC of three KT models with respect to the number of nodes. For each model, the AUC is obtained by averaging the results under the *cor* and *sum* aggregation settings. The dashed vertical line marks the beginning of the stable region, within which the configuration with $n = 2$ consistently achieves the best performance.

Further, we investigate the impact of different orders of spatial information (n=1,2,3) under varying numbers of spatial nodes, as shown in Fig. 3. The minimum number of spatial nodes is set to 7, which corresponds to the number of nodes contained within the third-order neighborhood (n=3) of a randomly selected target node. This choice ensures that spatial information of different orders is computed on a valid and comparable neighborhood scale. The maximum number of nodes is set to 25. In our experiments, when increasing the node number incrementally, model performance reaches its maximum and remains unchanged for five consecutive settings, after which further increases no longer yield observable gains. We therefore regard this point as the onset of a stable performance regime.

The results indicate that when the number of spatial nodes is small (e.g., 7 nodes), performance differences among different spatial orders are relatively unstable. In such cases, the relative ranking of different orders may vary across models or data splits, making it difficult to draw consistent conclusions.

As the number of spatial nodes gradually increases, this instability is substantially reduced. Once the node count reaches a certain scale, models consistently exhibit stable performance patterns across all experimental settings. In particular, models employing second-order spatial information (n=2) generally achieve the most favorable overall performance, especially in terms of AUC. These results suggest that, when sufficient spatial structure is available, second-order spatial information is more effective in capturing the key dependencies between a target node and its neighborhood, without introducing excessive redundant structure. Overall, the findings indicate that the choice of spatial order is closely related to the available node scale: when the node set is small, the advantages of higher-order information are not yet fully manifested; however, beyond a certain node threshold, second-order spatial information achieves an optimal balance between information sufficiency and noise control, leading to stable and consistent performance gains.

4.5 Interpretability of the Spatial Structure of Knowledge Components

Through the model's performance, we cannot directly pinpoint the exact reasons why second-order spatial information improves model performance. To investigate which factors have a more direct and influential impact on the model's performance, we conducted a detailed feature analysis using single-sample explanations (force plot), global importance, and feature density scatter plots.

We employed SHAP for this analysis, starting with single-sample explanations as shown in Figure 4. Features contributing positively to the prediction are shown in red, indicating an improvement in the prediction result, while features contributing negatively are shown in blue, indicating a decrease in the prediction result. The base_value represents the dataset's average prediction on the model. In Figure 4(a), the feature with the greatest impact on the prediction for the first sample is the total score of correct answers. This indicates that the more questions a learner answers correctly within the spatial network, the greater their mastery of the target KC. Meanwhile, the score of sibling nodes has the largest negative impact, suggesting that an inability

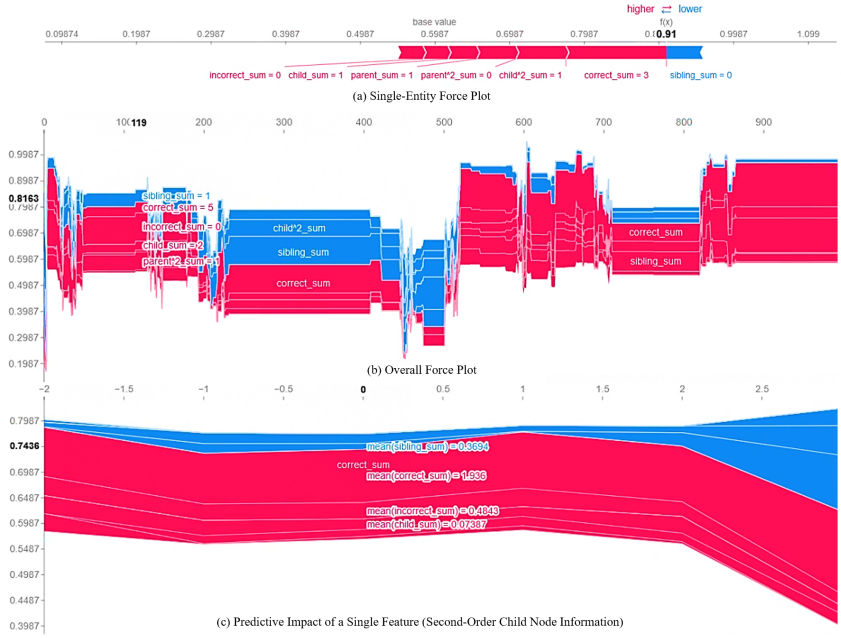

Fig. 4: SHAP-based explanations of the spatial KT model. (a) Single-entity force plot for an individual prediction. (b) Overall force plot aggregated across all samples. (c) Predictive impact of a single feature, using second-order child-node information as an example. Red and blue regions indicate positive and negative contributions to the prediction, respectively.

to correctly answer questions related to sibling KCs negatively affects the learner's performance on the target KC.

Based on this conclusion, we stacked the explanations of each single sample from the entire dataset, as shown in Figure 4(b). The figure demonstrates that while feature attributions vary slightly for each sample, the overall conclusion remains consistent with that of the single-sample explanation. In Figure 4(c), when selecting the scores of second-order child nodes as variables, different features have varying impacts on the prediction results.

On the other hand, through global interpretation, we aim to explain the overall structure of the model, which is more complex than explaining a single prediction. The goal is to illustrate feature importance across the entire dataset by averaging the SHAP values for each feature. In other words, the total score of correct answers, the score of sibling nodes, the score of incorrect answers, and the score of second-order child nodes are crucial for determining whether a learner has mastered the target KC and for predicting whether they can answer correctly, as shown in Figure 5(a).

Using the feature density scatter plot (beeswarm plot), we can see a concentrated summary of how second-order spatial features influence the model's output. In Figure 5(b), the x-axis represents the SHAP values, while the y-axis shows the features

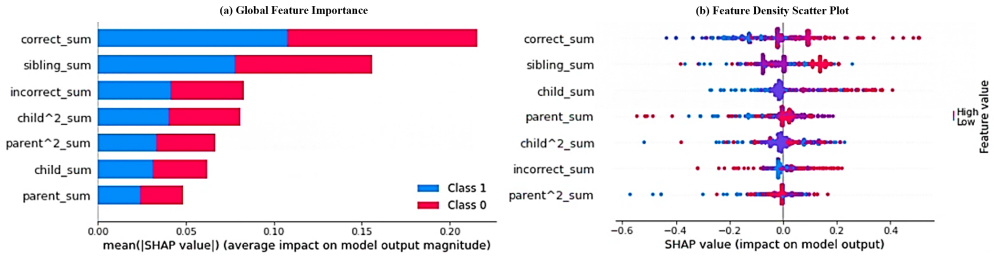

Fig. 5: Global explanation

included in the model, ranked by feature importance based on the mean SHAP values. It is evident that the most important feature in this model is correct_sum. Each dot in the plot represents a sample, and densely clustered areas along each row indicate regions with a large number of samples. In the plot, red represents higher feature values, while blue represents lower feature values. Taking correct_sum as an example, as its feature value increases, the SHAP value also increases, resulting in a positive impact. Conversely, for parentˆ2_sum, there is no clear correlation between its value and the SHAP value's direction or magnitude.

5 Discussion

This study explores various KT methods and models, focusing on the impact of incorporating multi-level spatial features and causal relationships between KCs to enhance model performance. The experimental results demonstrate that multi-level spatial structures have a significant effect on the predictive accuracy of the models.

First, the experiments show that integrating spatial information improves model performance across both deep learning models (such as DKT, DKVMN, and SAKT) and machine learning methods. This suggests that spatial relationships between KCs provide critical contextual information for predicting learners' mastery, especially when these relationships are organized hierarchically. Among the different models, those utilizing second-order spatial information consistently outperform models that rely solely on first-order or third-order structures. This may be because second-order nodes capture important dependencies that are not accessible to first-order nodes, while third-order nodes introduce redundant information, which negatively affects predictions.

Further analysis reveals that considering both correct and incorrect responses is crucial for predicting learner performance. Models using penalty mechanisms, where incorrect answers incur deductions, perform significantly better than those that consider only correct responses. This aligns with the theory that both successes and failures during the learning process provide valuable insights into a learner's knowledge state.

In terms of model interpretability, we conducted an in-depth analysis using SHAP values. Single-sample explanations show that the total score for correct answers has the most positive impact on predicting mastery of target knowledge, while poor performance on sibling nodes (i.e., KCs sharing the same parent node) negatively affects the

prediction. This indicates that mastering related KCs is critical for successfully answering questions related to the target KC. Global feature importance analysis further confirms that the total score for correct answers, sibling node scores, and second-order child node scores are the most important features in predicting learner performance, highlighting the importance of multi-level spatial structures in KT.

However, the complexity introduced by higher-order spatial features and causal relationships also poses challenges in terms of model interpretability and computational efficiency. While second-order structures perform best, the added complexity of third-order structures often provides diminishing or even negative returns. Moreover, models based on deep learning architectures, such as DKVMN and SAKT, although highly effective, tend to lack interpretability, making it difficult to fully understand their decision-making processes. The issue of limited interpretability is particularly critical in educational applications, where the trustworthiness of model outputs is of utmost importance.

6 Limitations

Despite demonstrating the potential of multi-level spatial features and causally informed relationships to improve the performance of KT models, this study also has several limitations that call for further investigation. First, the complexity of the models increases substantially with the introduction of multi-level spatial features, particularly when second-order and third-order nodes are incorporated. This not only lengthens training time but also increases computational and memory demands, which may hinder deployment in real-world educational settings, especially in resource-constrained environments. A more compact representation of spatial information, or adaptive mechanisms that select only the most informative neighbors, would be valuable directions for future work.

Second, a key limitation lies in our use of fGES to infer a causal structure over KCs from purely observational student response data. Although we refer to the resulting graph as a causal structure and ground it in standard assumptions, it does not establish definitive causal relations among KCs. Score-based causal discovery methods such as fGES assume causal sufficiency, acyclicity, and faithfulness; however, these assumptions are only approximations in educational settings where many relevant factors remain unobserved. For example, latent general ability, motivation, classroom context, teacher practices, and curriculum design can act as unmeasured confounders that simultaneously affect multiple KCs. Such unobserved confounders may distort the learned graph in different ways: they can introduce spurious edges between KCs that tend to be taught together or are similarly influenced by latent traits, and they can mask or weaken direct prerequisite relations when strong confounders dominate the observed associations. These distortions can in turn affect our knowledge transfer predictions and the interpretation of spatial features. While our experiments show that incorporating the learned structure improves KT performance, individual edges should therefore be viewed as hypotheses about potential skill dependencies rather than as confirmed instructional prerequisites.

Future work could mitigate these limitations in several ways. One direction is to combine fGES with prior domain knowledge and curriculum constraints, for example by constraining or regularizing the search space of graphs, or by pruning edges that are inconsistent with expert-designed learning trajectories. Another is to incorporate interventional or quasi-experimental data (e.g., targeted practice or instructional interventions on specific KCs) to validate and refine the learned structure. It would also be valuable to explore causal discovery methods that explicitly account for latent confounders, as well as stability analyses (e.g., bootstrapping the data) to assess the robustness of individual edges. These extensions could lead to more reliable causal graphs and, in turn, more trustworthy spatial features for KT models.

7 Conclusion

This study investigates the impact of spatial structures on improving the performance of KT models. The experimental results show that incorporating spatial structure information, particularly second-order spatial structures, significantly enhances the predictive accuracy of KT models. Compared with models without spatial structure information, models using second-order spatial features generally achieved better predictive performance, with more evident gains in AUC than in accuracy. The improvement was particularly pronounced for the interpretable SpaKT model, while the gains for deep KT models were relatively modest but generally positive. This indicates that hierarchical dependencies between KCs play a critical role in predicting learners' knowledge states. The second-order spatial features effectively capture the complex relationships between KCs, enabling the model to more accurately assess learners' mastery of target KCs. Additionally, integrating both correct and incorrect response data further enhances the model's predictive capabilities, resulting in more stable performance.

Overall, this study highlights the importance of different levels of spatial structures in KT models, demonstrating that optimizing the network of KC relationships can significantly improve model performance. These findings provide valuable insights for future improvements in KT methodologies.

8 Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 62477012), the Natural Science Foundation of Shanghai (Grant No. 23ZR1418500), and the AI for Science Program of the Shanghai Municipal Commission of Economy and Informatization (Grant No. 2025-GZL-RGZN-BTBX-01014).

References

Abdelrahman, G., Wang, Q., Nunes, B. (2023). Knowledge tracing: A survey. *ACM Computing Surveys*, *55*(11), 1–37, https://doi.org/10.1145/3569576

Cen, H., Koedinger, K., Junker, B. (2006). Learning factors analysis–a general method for cognitive model evaluation and improvement. *International conference on intelligent tutoring systems* (pp. 164–175).

Cen, H., Koedinger, K., Junker, B. (2008). Comparing two irt models for conjunctive skills. *Intelligent tutoring systems: 9th international conference, its 2008, montreal, canada, june 23-27, 2008 proceedings 9* (pp. 796–798).

Chaiklin, S. (2003). The zone of proximal development in vygotsky's analysis of learning and instruction. *Vygotsky's educational theory in cultural context* (Vol. 1, pp. 39–64). Cambridge, UK: Cambridge University Press.

Chounta, I.-A., Albacete, P., Jordan, P., Katz, S., McLaren, B.M. (2017). The "grey area": A computational approach to model the zone of proximal development. *Data driven approaches in digital education: 12th european conference on technology enhanced learning, ec-tel 2017, tallinn, estonia, september 12–15, 2017, proceedings 12* (pp. 3–16).

Corbett, A.T., & Anderson, J.R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, *4*, 253–278, https://doi.org/10.1007/BF01099821

Gan, W., Sun, Y., Peng, X., Sun, Y. (2020). Modeling learner's dynamic knowledge construction procedure and cognitive item difficulty for knowledge tracing. *Applied Intelligence*, *50*, 3894–3912, https://doi.org/10.1007/s10489-020-01756-7

Jiang, B., Wei, Y., Zhang, T., Zhang, W. (2024). Improving the performance and explainability of knowledge tracing via markov blanket. *Information Processing & Management*, *61*(3), 103620, https://doi.org/10.1016/j.ipm.2023.103620

Liu, Q., Huang, Z., Yin, Y., Chen, E., Xiong, H., Su, Y., Hu, G. (2019). Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering*, *33*(1), 100–115, https://doi.org/10.1109/TKDE.2019.2924374

Lundberg, S.M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems* (Vol. 30, pp. 1–10).

Nagatani, K., Zhang, Q., Sato, M., Chen, Y.-Y., Chen, F., Ohkuma, T. (2019). Augmenting knowledge tracing by considering forgetting behavior. *The world wide web conference* (pp. 3101–3107).

Nakagawa, H., Iwasawa, Y., Matsuo, Y. (2019). Graph-based knowledge tracing: modeling student proficiency using graph neural network. *Ieee/wic/acm international conference on web intelligence* (pp. 156–163).

Nnadi, L.C., Watanobe, Y., Rahman, M.M., John-Otumu, A.M. (2024). Prediction of students' adaptability using explainable ai in educational machine learning models. *Applied Sciences*, *14*(12), 5141, https://doi.org/10.3390/app14125141

Pandey, S., & Karypis, G. (2019). A self-attentive model for knowledge tracing. *International conference on education data mining* (pp. 1–6).

Pavlik, P.I., Cen, H., Koedinger, K.R. (2009). Performance factors analysis–a new alternative to knowledge tracing. *Artificial intelligence in education* (pp. 531–538).

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L.J., Sohl-Dickstein, J. (2015). Deep knowledge tracing. *Advances in neural information processing systems* (Vol. 28, pp. 1–9).

Ritter, S., Yudelson, M., Fancsali, S.E., Berman, S.R. (2016). How mastery learning works at scale. *Proceedings of the third (2016) acm conference on learning@ scale* (pp. 71–79).

Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G. (2008). The graph neural network model. *IEEE transactions on neural networks*, *20*(1), 61–80, https://doi.org/10.1109/TNN.2008.2005605

Shen, S., Liu, Q., Chen, E., Wu, H., Huang, Z., Zhao, W., ... Wang, S. (2020). Convolutional knowledge tracing: Modeling individualization in student learning process. *Proceedings of the 43rd international acm sigir conference on research and development in information retrieval* (pp. 1857–1860).

Sun, X., Liu, Q., Zhang, K., Shen, S., Yang, L., Li, H. (2025). Harnessing code domain insights: Enhancing programming knowledge tracing with large language models. *Knowledge-Based Systems*, *317*, 113396, https://doi.org/10.1016/j.knosys.2025.113396

Sun, X., Zhang, K., Liu, Q., Shen, S., Wang, F., Guo, Y., Chen, E. (2025). Daskt: A dynamic affect simulation method for knowledge tracing. *IEEE Transactions on Knowledge and Data Engineering*, *37*(4), 1714–1727, https://doi.org/10.1109/TKDE.2025.3526584

Tong, S., Liu, Q., Huang, W., Huang, Z., Chen, E., Liu, C., ... Wang, S. (2020). Structure-based knowledge tracing: An influence propagation view. *2020 ieee international conference on data mining (icdm)* (pp. 541–550).

Vie, J.-J., & Kashima, H. (2019). Knowledge tracing machines: Factorization machines for knowledge tracing. *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 750–757).

Wei, Y., & Jiang, B. (2023). Interpretable cognitive state prediction via temporal fuzzy cognitive map. *IEEE Transactions on Learning Technologies*, *17*, 514–526, https://doi.org/10.1109/TLT.2023.3307565

Wei, Y., Zhou, Y., Jiang, Y.-H., Jiang, B. (2024). Enhancing explainability of knowledge learning paths: causal knowledge networks. *International conference on educational data mining* (pp. 1–8).

Wilson, K.H., Karklin, Y., Han, B., Ekanadham, C. (2016). Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation. *International conference on educational data mining* (pp. 1–6).

Yang, Y., Shen, J., Qu, Y., Liu, Y., Wang, K., Zhu, Y., ... Yu, Y. (2021). Gikt: a graph-based interaction model for knowledge tracing. *Machine learning and knowledge discovery in databases: European conference, ecml pkdd 2020, ghent, belgium, september 14–18, 2020, proceedings, part i* (pp. 299–315).

Yudelson, M.V., Koedinger, K.R., Gordon, G.J. (2013). Individualized bayesian knowledge tracing models. *Artificial intelligence in education: 16th international conference, aied 2013, memphis, tn, usa, july 9-13, 2013. proceedings 16* (pp. 171–180).

Zhang, J., Shi, X., King, I., Yeung, D.-Y. (2017). Dynamic key-value memory networks for knowledge tracing. *Proceedings of the 26th international conference on world wide web* (pp. 765–774).

Zhang, M., Zhu, X., Zhang, C., Qian, W., Pan, F., Zhao, H. (2023). Counterfactual monotonic knowledge tracing for assessing students' dynamic mastery of knowledge concepts. *Proceedings of the 32nd acm international conference on information and knowledge management* (pp. 3236–3246).