

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

© The Authors 2026. This article is published with open access at link.springer.com and journal.hep.com.cn

Abstract Knowledge tracing (KT) predicts learners' evolving knowledge states by tracking their performance over time. While temporal dynamics have been the focus of most KT studies, spatial structures among knowledge components (KCs) remain underexplored, despite containing rich latent information. Prior work suggested that inter-KCs relationships can enhance KT performance, yet the impact of hierarchical spatial structures remains unclear. This study investigates how multilevel spatial relationships among KCs affect the performance of KT 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 KT models. Experimental results showed that incorporating second-order spatial structures yielded consistent performance gains. These findings underscore the value of spatial structural information in KT. Furthermore, interpretable feature analyses illustrated how spatial features shape diagnostic predictions, providing insight into factors underlying students' learning challenges. This spatial perspective not only improves KT models' 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 et al., 2016). A critical aspect of this identification is understanding the features and factors that influence learners' mastery of KCs, as these are key to measuring cognitive states precisely (Zhang et al., 2023). Existing work in cognitive diagnosis, particularly in knowledge tracing (KT), focused mainly 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, 2024). In addition, some studies modeled relational structures among KCs to extract behavioral associations (Nakagawa et al., 2019; Wei et al., 2024; Yang et al., 2021). Despite these advances, the multilevel, hierarchical organization of KCs remains underexplored. In this study, we refer to this type of 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 et al., 2023; Jiang et al., 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.

Received November 14, 2025; revised December 28, 2025; accepted May 22, 2026

Bo Jiang (✉)

E-mail: bjiang@deit.ecnu.edu.cn

‡ These authors contributed equally.

Beyond predictive performance, interpretability is essential. Whether identified features genuinely affect mastery of a target KC, which features exert the strongest influence on assessment of that target, and why they have such effects are key questions for building teachers' and students' trust 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-, second-, and third-order nodes, we intend to capture multilevel dependencies among KCs and integrate the resulting spatial structures into cognitive diagnostic models. Experimental results showed that leveraging these structures improved diagnostic performance. In particular, second-order spatial features yielded the most favorable overall results, with the largest improvements observed in the interpretable feature-based model and more modest but generally positive gains observed in deep KT models. These findings highlight the important role of hierarchical dependencies among KCs in learners' mastery of target concepts. The findings further suggest that effective instructional interventions may benefit from identifying the second-order parent nodes of the target 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 analysis methods based on Shapley additive explanations (SHAP) (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 observed the contributions of different features and identified which KCs within the spatial structure were 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, this study makes three main contributions:

(1) This study systematically investigates how hierarchical knowledge structures affect learning outcomes within a cognitive diagnosis framework;

(2) This study proposes a transparent and interpretable KT framework that can be instantiated with a variety of KT models;

(3) This study demonstrates 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 sequences of correct and incorrect responses (Abdelrahman et al., 2023). Essentially, KT involves dynamic tracing of learners' knowledge states, and recent studies have further extended this process by considering affective dynamics during learning (Sun et al., 2025b). KT models are typically divided into two categories: traditional KT models and deep KT models.

Early KT studies mainly relied on probabilistic models, such as Bayesian knowledge tracing (BKT) model (Corbett & Anderson, 1994) and its personalized variants (Yudelson et al., 2013), as well as IRT- and factor-analysis-based models, including HIRT (Wilson et al., 2016), AFM (Cen et al., 2008), and PFA (Pavlik et al., 2009). Recent studies have increasingly adopted deep-learning architectures, including DKT (Piech et al., 2015), Dynamic key-value memory networks (DKVMN) (Zhang et al., 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 for improving KT performance: knowledge graph-based modeling, knowledge structure modeling, and feature selection (Sun et al., 2025a). Knowledge graph-based methods introduce graph structures into KT; for example, GNNs (Scarselli et al., 2009) and models such as GIKT (Yang et al., 2021) are used to capture dependencies between KCs and interactions between questions and knowledge elements, although 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., 2021), 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 et al., 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 of 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, limiting their transparency and trustworthiness in practical applications.

2.2 | Explainable Knowledge Tracing Models

Existing KT methods have made significant progress in terms of explainability. However, these efforts are often limited to specific model structures, lacking a universally applicable framework for explanation. For instance, the 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, the BKT model overlooks long-term dependencies and forgetting phenomena, limiting its ability to capture complex learning behaviors. Similarly, learning factor analysis (Cen et al., 2006) is easy to explain, but it assumes a uniform learning rate across all learners and KCs, thus lacking adaptability to individual learning patterns.

More complex models, such as DKVMN and convolutional knowledge tracing (CKT) (Shen et al., 2020), can model forgetting and capture interactions between learners and KCs, but their complexity and long training times make them difficult to deploy in practical settings. In addition, 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) offers an innovative game-theoretic approach designed to explain model outputs and 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. (2024) applied SHAP to predict learners' adaptability. Their study revealed that factors such as "class duration" and "financial condition" significantly influence learners' adaptability. 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.

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

3 Methodology

3.1 | Knowledge Component Network Structure

To investigate the impact of hierarchical dependencies among KCs on KT methods, we first define how KCs are connected, by modeling the KC network as a directed graph, with each KC a node and each prerequisite or causal relation a directed edge between two nodes. In Figure 1, the distance between two KCs is defined as the number of edges on the shortest path connecting them.

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

In addition, considering the multilevel influence between KCs, we categorize the structural space into first-, second-, 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. In a strict graph-theoretic sense, sibling nodes are not necessarily one edge away from the target node, yet they are treated as immediate pedagogically related nodes, given that they share the same parent as the target KC. Following the work on the zone of proximal development in educational psychology and its computational operationalizations (Chaiklin, 2003; Chounta et al., 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, shown in Figure 1(b).

The set of parent nodes (N_{parents}), child nodes (N_{children}), and sibling nodes (N_{siblings}) is defined as the first-order immediate nodes ($N_{\text{First-Im}}$). Parent and child nodes are directly connected to the target KC, while

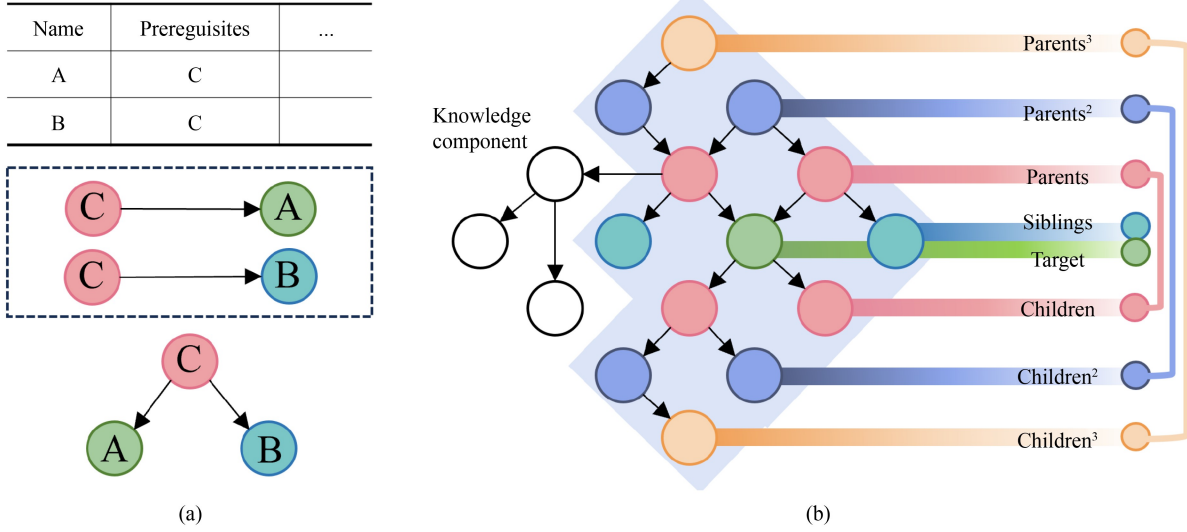


Figure 1 Multilevel spatial structure of knowledge components (KCs). (a) Basic KC relationships; (b) multilevel spatial structure of a target KC.

sibling nodes are included as they share an immediate parent with the target KC:

$$N_{\text{First-Im}} \triangleq \{N_{\text{parents}}, N_{\text{children}}, N_{\text{siblings}}\}. \quad (1)$$

The collection of parent nodes, child nodes, sibling nodes, second-order parent nodes (N_{parents^2}), and second-order child nodes (N_{children^2}) of the target KC are defined as second-order immediate nodes ($N_{\text{Second-Im}}$):

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

The collection of the parent nodes, child nodes, sibling nodes, second-order parent nodes, second-order child nodes, third-order parent nodes (N_{parents^3}), and third-order child nodes (N_{children^3}) of the target KC are defined as third-order immediate nodes ($N_{\text{Third-Im}}$):

$$N_{\text{Third-Im}} \triangleq \left\{ \begin{array}{c} N_{\text{parents}}, N_{\text{parents}^2}, N_{\text{parents}^3}, \\ N_{\text{children}}, N_{\text{children}^2}, N_{\text{children}^3}, N_{\text{siblings}} \end{array} \right\}. \quad (3)$$

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 $\mathcal{K} = \{K_1, K_2, \dots, K_n\}$, where each KC K_i corresponds to a random variable. The goal is to learn a directed acyclic graph $G = (V, E)$ from the observed data $\mathbf{D}$, where V represents the set of nodes corresponding to $\mathcal{K}$; 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:

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

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

To compute the graph, we begin 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 the edge and compute the score $S(G', \mathbf{D})$ for the new graph G' formed by its addition. 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.

Next, for each possible edge ($K_i \rightarrow K_j$) 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.

Finally, 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 Algorithm 1 to infer a directed graph over KCs from observational student response data. Following standard causal discovery practice, fGES assumes:

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

```

1:  $G \leftarrow G_0$ 
2: Forward Phase:
3: while true do
4:    $G_{\text{best}} \leftarrow G$ 
5:   for each edge  $(K_i \rightarrow K_j)$  not in  $G$  do
6:      $G' \leftarrow G \cup \{K_i \rightarrow K_j\}$ 
7:     if  $S(G', \mathbf{D}) > S(G_{\text{best}}, \mathbf{D})$  then
8:        $G_{\text{best}} \leftarrow G'$ 
9:     end if
10:   end for
11:   if  $G_{\text{best}} = G$  then
12:     break
13:   else
14:      $G \leftarrow G_{\text{best}}$ 
15:   end if
16: end while
17: Backward Phase:
18: while true do
19:    $G_{\text{best}} \leftarrow G$ 
20:   for each edge  $(K_i \rightarrow K_j)$  in  $G$  do
21:      $G' \leftarrow G \setminus \{K_i \rightarrow K_j\}$ 
22:     if  $S(G', \mathbf{D}) > S(G_{\text{best}}, \mathbf{D})$  then
23:        $G_{\text{best}} \leftarrow G'$ 
24:     end if
25:   end for
26:   if  $G_{\text{best}} = G$  then
27:     break
28:   else
29:      $G \leftarrow G_{\text{best}}$ 
30:   end if
31: end while
32: return  $G$ 

```

Note. fGES denotes fast Greedy Equivalence Search.

- (1) Causal sufficiency (no unmeasured common causes among the observed variables);
- (2) Acyclicity of the underlying causal graph;
- (3) Causal Markov and faithfulness conditions;
- (4) 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 from direct

prerequisite relations, and some true dependencies may be missed. Therefore, we use the fGES-based structure primarily as a causally informed representation to construct spatial features for KT in this study, and we are cautious about drawing strong causal conclusions from individual edges.

3.2 | Knowledge Tracing Model

We consider two feature-encoding schemes regarding the information of related KCs (Table 1). Under Assumption I, for a related KC, each correct response 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 accumulates only successful responses. Under Assumption II, for a related KC, each correct response 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 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.

Table 1 Features under the two assumptions

Assumption	Feature	Description
I	<i>children_corrects</i>	The correct score of the child nodes
	<i>parents_corrects</i>	The correct score of the parent nodes
	<i>siblings_corrects</i>	The correct score of the sibling nodes
II	<i>children_sum</i>	The score of the child nodes
	<i>parents_sum</i>	The score of the parent nodes
	<i>siblings_sum</i>	The score of the sibling nodes
	<i>correct_sum</i>	The total score for correct answers
	<i>incorrect_sum</i>	The total score for incorrect answers

Subsequently, using the causal relationship structure between KCs obtained in Subsection 3.1, as well as the multilevel KC information affecting the target nodes K_t , we incorporate additional information into the deep KT model $\mathcal{M}$. This step examines whether additional information enhances the predictive performance of the deep KT model. The detailed procedure is shown in Algorithm 2.

Deep learning models can achieve better accuracy in combination; 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. For example, we use decision trees, and the algorithm process is shown in Algorithm 3.

Algorithm 2 Integrating causal structures and multilevel knowledge into the deep knowledge tracing model

Require: Data $\mathbf{D}$, Causal Graph G , Target Node K_t , Deep Knowledge Tracing Model $\mathcal{M}$

Ensure: Updated Model $\mathcal{M}'$

- 1: **Step 1: Extract Knowledge**
 - 2: Identify ancestors of K_t in G and their levels
 - 3: **Step 2: Encode Knowledge**
 - 4: Construct feature vectors f_t incorporating causal and multilevel info
 - 5: $f_t \leftarrow \text{Encode}(G, \text{Multilevel})$
 - 6: **Step 3: Integrate with Model**
 - 7: Update input layer of $\mathcal{M}$ to include f_t
 - 8: Augment dataset: $\mathbf{D}' \leftarrow (\mathbf{D}, f_t)$
 - 9: Train $\mathcal{M}$ on $\mathbf{D}'$
 - 10: **Step 4: Optimize Model**
 - 11: Optimize $\mathcal{M}$ with backpropagation
 - 12: Evaluate and fine-tune $\mathcal{M}$
 - 13: **return** Updated model $\mathcal{M}'$
-

Algorithm 3 Constructing a decision tree model with causal and multilevel knowledge

Require: Dataset $\mathbf{D}$ with features $\mathbf{X}$, Target variable Y , Causal Graph G , Target Node K_t

Ensure: Decision Tree Model $\mathcal{T}'$

- 1: **Step 1: Extract Knowledge**
 - 2: Identify parents and higher-order ancestors of K_t in G
 - 3: Group them into sets $\{A_1, A_2, \dots, A_m\}$
 - 4: **Step 2: Encode Information**
 - 5: Create features f_t from ancestor data
 - 6: Augment dataset: $\mathbf{D}' \leftarrow (\mathbf{X}, f_t, Y)$
 - 7: **Step 3: Prepare Features**
 - 8: Normalize or standardize features f_t
 - 9: Optionally select relevant features
 - 10: **Step 4: Train Model**
 - 11: Initialize and train decision tree $\mathcal{T}$ using $\mathbf{D}'$
 - 12: **Step 5: Evaluate**
 - 13: Assess model performance
 - 14: Optimize hyperparameters
 - 15: **Step 6: Deploy**
 - 16: Analyze and deploy trained model $\mathcal{T}'$
 - 17: **return** Decision Tree Model $\mathcal{T}'$
-

4 Experiments

4.1 | Dataset

This study used the Junyi2015 dataset for experiments. Compared to other publicly available datasets, such as ASSISTments 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.

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. In addition, the *Junyi_Exercise_table* records detailed information about 819 questions. Attributes related to 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 on the knowledge map. The specific relationships are shown in Table 2.

Table 2 Feature structure of the Junyi2015 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

The data used in this study should meet two basic requirements:

- (1) The selected KCs must have corresponding answer records;
- (2) The answer records should be sufficiently numerous.

After performing statistical filtering on the dataset, we selected data from the “geometry” area, obtained from the *Junyi_Exercise_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 matched. These KCs were then removed, resulting in a final dataset containing 2,922,789 answer records.

4.2 | Experimental Setup

For all deep KT 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 5 GB 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 had substantially lower computational costs than deep KT models, as it does not involve iterative backpropagation. In our experiments, SpaKT training was 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 achieved consistent performance improvements with acceptable computational costs.

4.3 | Evaluation Metrics

This study constructed models under different KC scenarios and compared them with baseline models to evaluate their effectiveness. Three commonly used metrics, namely accuracy, area under the curve (AUC), and root mean square error (RMSE), were adopted.

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

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}, \quad (5)$$

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

AUC represents the area under the Receiver Operating Characteristic (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:

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

where N denotes the total number of samples, y_i is the actual value of the i -th sample, and $\hat{y}_i$ is the corresponding predicted value.

4.4 | Impact of Spatial Structure on Knowledge Tracing Performance

To validate the impact of multilevel 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. For example, we used *congruent_triangles_1* as the central KC to demonstrate how subgraphs are selected. The causal relationships between the KCs in the subgraph were obtained using the method outlined in Subection 3.1, and the resulting causal structure was used for KT experiments.

Figure 2 presents the knowledge space structure diagram centered on the KC *congruent_triangles_1*. The original relationship diagram (see Figure 2(a)) as well as its corresponding causal structure diagram (see Figure 2(b)) are also uploaded in the Electronic Supplementary Material. We used 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 shown in Figure 2(d), we summarized 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) represents the changes in node relationships from the prerequisite relationship diagram to the causal diagram.

Based on the causal structure learning procedure and the multilevel spatial feature construction described in Subsections 3.1 and 3.2, we incorporated the resulting KC information into both the deep learning and machine learning KT models. The corresponding performance comparisons are reported in Tables 4 and 5.

Table 4 presents the performance of models using raw information, such as question IDs and answer sequences, while Table 5 shows the performance of models incorporating spatial feature information. A comparison revealed 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 achieved a better AUC

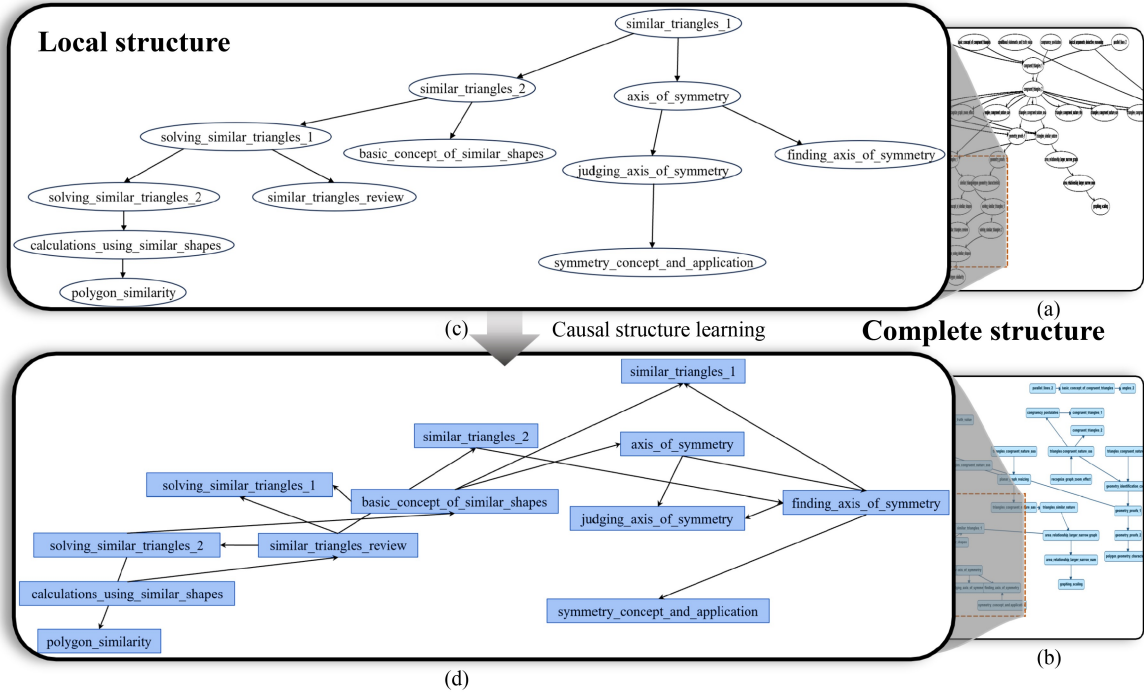


Figure 2 Causal graph structure of the knowledge components. (a) Original prerequisite relationship diagram of the knowledge components; (b) causal structure diagram learned from the data, using (a) as the initial graph. (c) and (d) are respectively extracted sections of (a) and (b) for clearer viewing.

Table 3 Comparison of causal relationships and original prerequisite relationships

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

after incorporating second-order spatial information, although the improvements were not uniform across all metrics. In addition, for SpaKT, compared to Assumption I in Table 1, which does not include penalty features, Assumption II in Table 1, which incorporates penalty features, achieved a 0.5% increase in AUC.

Based on these results, we conclude the following:

(1) Spatial information can enhance the model's performance to some extent;

(2) Using second-order spatial information resulted in the best model performance;

(3) Feature processing with penalties led to better model performance.

First, the improvement brought by spatial information was validated in both deep learning and

machine learning methods. Second, second-order spatial information likely led to the best model performance because second-order nodes contained critical information that was not present in first-order nodes, whereas third-order nodes introduced some redundant information, which negatively impacted the prediction of target nodes. Third, in the process of identifying students' responses to questions, incorrect answers also contributed to prediction accuracy. Therefore, incorporating both correct and incorrect response statistics performed better than using correct information alone.

Furthermore, we investigated the impact of different orders of spatial information ($n = 1, 2$, and 3) under varying numbers of spatial nodes, as shown in Figure 3. The minimum number of spatial nodes was set to 7, corresponding to the number of nodes contained within the third-order neighborhood ($n = 3$) of a randomly selected target node. This choice ensured that spatial information of different orders was computed on a valid and comparable neighborhood scale. The maximum number of nodes was set to 25. In our experiments, as the number of nodes increased incrementally, the model performance reached its maximum and remained unchanged for 5 consecutive settings, after which further increases no longer yielded observable gains. We therefore regard this point as the onset of a stable performance regime.

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

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

As the number of spatial nodes gradually increases, the instability is substantially reduced. Once the node count reaches a certain scale, the 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,

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

Notes. 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 Assumption I in Table 1, while *sum* refers to using the features derived from Assumption II in Table 1. The values in bold represent the best performance metrics within the group.

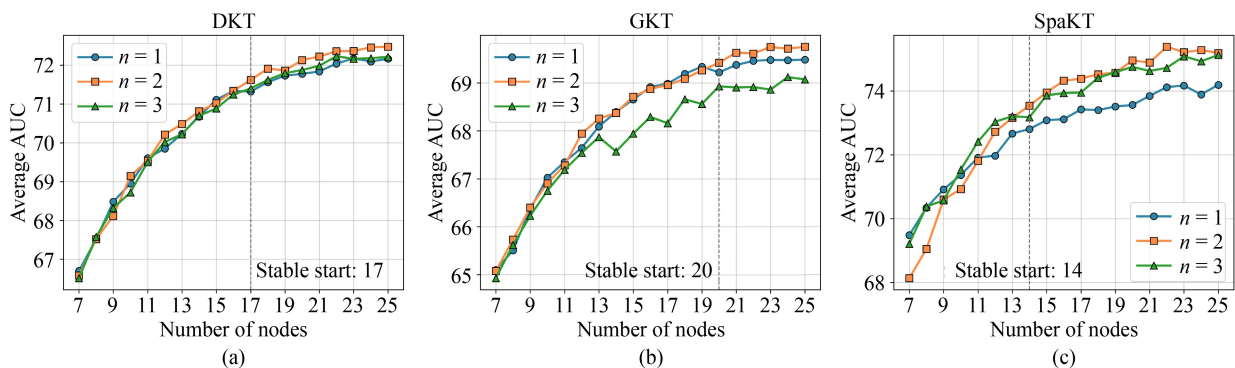


Figure 3 Average AUC of three KT models with respect to the number of nodes. (a) DKT; (b) GKT; (c) SpaKT. For each model, the AUC was obtained by averaging the results under the *cor* and *sum* aggregation settings. A dashed vertical line marks the beginning of the stable region, within which the configuration with $n = 2$ consistently achieved the best 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 inevident; however, beyond a certain node threshold, second-order spatial information achieve 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

Based on the model’s performance, we cannot directly pinpoint the exact reasons why second-order spatial information improved model performance. To investigate which factors have more direct and influential impacts on the model’s performance, we conducted a

detailed feature analysis using single-sample explanations (force plot), global importance, and feature-density scatter plots.

SHAP was employed 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 answered correctly within the spatial network, the greater his mastery of the target KC. Meanwhile, the score of sibling nodes had the largest negative impact, suggesting that an inability to correctly answer questions related to sibling KCs negatively affected the learner’s performance on the target KC.

Based on this finding, we stacked the explanations for individual samples across the entire dataset, as

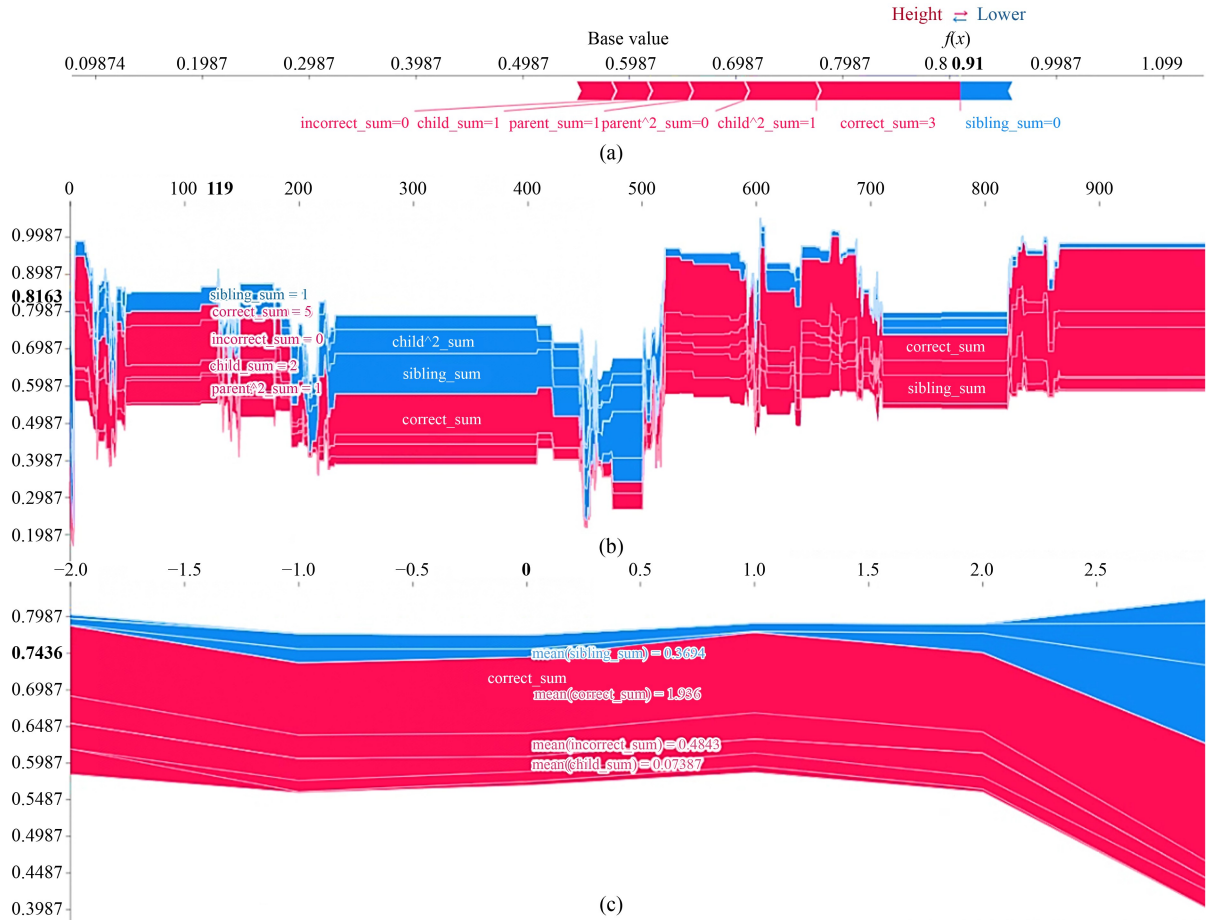


Figure 4 SHAP-based explanations of the spatial KT model. (a) Single-entity force plot for an individual prediction; (b) overall force plot aggregated across all samples; (c) predictive impact of a single feature, using second-order child node information as an example. The red and blue regions indicate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

shown in Figure 4(b). The figure demonstrates that while feature attributions varied slightly for each sample, the overall conclusion remained 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 had varying impacts on the prediction results.

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 a 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 included in the model, rank by feature importance based on the mean SHAP values. 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 *parent^2_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 multilevel

spatial features and causal relationships between KCs to enhance model performance. The experimental results showed that multilevel spatial structures had a significant effect on the predictive accuracy of the models.

First of all, the experiments showed that integrating spatial information improved model performance across both deep learning models (e.g., 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 outperformed models relying solely on first- 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 revealed that considering both correct and incorrect responses was crucial for predicting learner performance. Models using penalty mechanisms, in which incorrect answers incur deductions, performed significantly better than those that considered only correct responses. This aligns with the theory that successes and failures during the learning process provide valuable insights into a learner's knowledge state.

In terms of model explainability, we conducted an in-depth analysis using SHAP values. Single-sample explanations showed that the total score for correct answers had 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 affected predictions. This indicates that mastering related KCs is critical for successfully answering questions related to the target KC. Global feature importance analysis further confirmed that the total score for correct answers, sibling node scores, and second-order child node scores were the most important features in predicting learner performance,

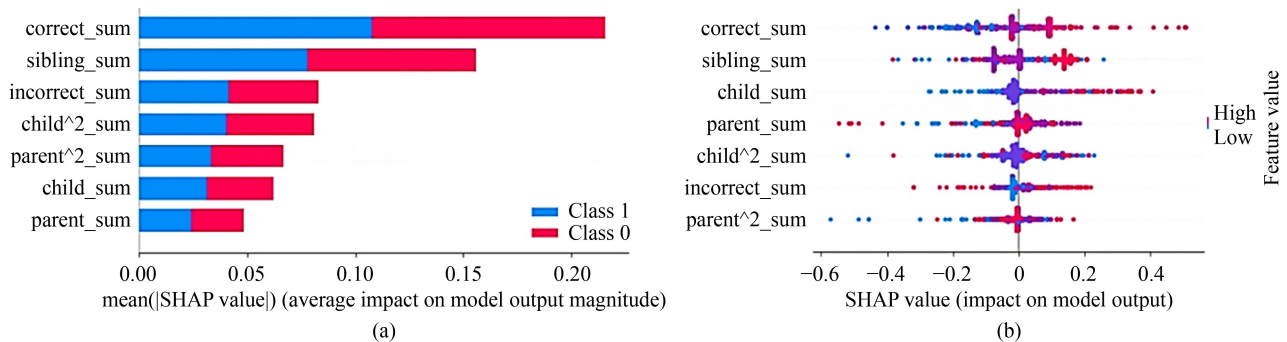


Figure 5 Global explanation. (a) Global feature importance; (b) feature density scatter plot. In (a), Class 1 and Class 0 denote correct and incorrect response outcomes, respectively. In (b), red and blue indicate high and low feature values, respectively.

highlighting the importance of multilevel 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 performed best, the added complexity of third-order structures often provided diminishing or even negative returns. Moreover, models based on deep learning architectures, such as DKVMN and SAKT, although highly effective, tended to lack interpretability, making it difficult to fully understand their decision-making processes. Limited explainability is particularly critical in educational applications, where the trustworthiness of model outputs is essential.

6 Limitations

Despite demonstrating the potential of multilevel spatial features and causally informed relationships to improve the performance of KT models, this study has two limitations that call for further investigation. The first one is the complexity of the models increases substantially with the introduction of multilevel 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, potentially hindering deployment in real-world educational settings, especially in resource-constrained environments. More compact representations of spatial information or adaptive mechanisms that select only the most informative neighbors would be valuable directions for future work.

The second one lies in the 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 by introducing spurious edges between KCs that are frequently taught together or influenced by the same latent traits, while masking or weakening direct prerequisite relations when confounding effects dominate the observed associations. These distortions can, in turn, affect our knowledge transfer predictions and the interpretation of spatial features.

While our experiments showed 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 two limitations in several ways. One possible 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, such as 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, such as 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 Conclusions

This study investigates the impact of spatial structures on improving the performance of KT models. The experimental results showed that incorporating spatial structure information, particularly second-order spatial structures, significantly enhanced 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 explainable SpaKT model, while the gains for deep KT models were relatively modest but generally positive. These findings indicate 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. In addition, integrating both correct and incorrect response data can further enhance 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 improving future KT methodologies.

Acknowledgments This study 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).

Author Contributions Yuang Wei was responsible for conceptualization, methodology, project administration, formal analysis, and manuscript drafting. Rui Jia undertook data curation, formal analysis, investigation, validation, and original draft composition. Yingwen Ding contributed to conceptualization, investigation, and drafting of the original manuscript. Bo Jiang provided supervision, secured research funding, and oversaw the manuscript drafting and review.

Conflict of Interest The authors declare no competing interests.

Data Availability Statements The code is available at <https://github.com/Assore/XKT>.

Electronic Supplementary Material Supplementary material is available in the online version of this article at <https://doi.org/10.1007/s44366-026-0097-8> and is accessible for authorized users.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution, and reproduction in any medium or format as long as appropriate credit is given to the original author(s) and source, a link to the Creative Commons license is provided, and the changes made are indicated. The images or other third-party material in this article are included in the article's Creative Commons license unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. Visit <http://creativecommons.org/licenses/by/4.0/> to view a copy of this license.

References

- Abdelrahman, G., Wang, Q., & Nunes, B. (2023). Knowledge tracing: A survey. *ACM Computing Surveys*, 55(11), 224.
- Cen, H., Koedinger, K., & Junker, B. (2006). Learning factors analysis—A general method for cognitive model evaluation and improvement. In: *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*. Taoyuan: Springer, 164–175.
- Cen, H., Koedinger, K., & Junker, B. (2008). Comparing two IRT models for conjunctive skills. In: *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*. Montreal: Springer, 796–798.
- Chaiklin, S. (2003). The zone of proximal development in Vygotsky's analysis of learning and instruction. In: Kozulin, A., Gindis, B., Ageyev, V. S., & Miller, S. M., eds. *Vygotsky's educational theory in cultural context*. Cambridge: Cambridge University Press, 39–64.
- 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. In: *Proceedings of the 12th European Conference on Technology Enhanced Learning Data Driven Approaches in Digital Education*. Tallinn: Springer, 3–16.
- Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4), 253–278.
- Gan, W. B., Sun, Y., Peng, X., & Sun, Y. (2020). Modeling learner's dynamic knowledge construction procedure and cognitive item difficulty for knowledge tracing. *Applied Intelligence*, 50(11), 3894–3912.
- Jiang, B., Wei, Y. A., Zhang, T., & Zhang, W. (2024). Improving the performance and explainability of knowledge tracing via Markov blanket. *Information Processing & Management*, 61(3), 103620.
- Liu, Q., Huang, Z. Y., Yin, Y., Chen, E. H., Xiong, H., Su, Y., & Hu, G. P. (2021). EKT: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering*, 33(1), 100–115.
- Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach: Curran Associates Inc., 1–10.
- Nagatani, K., Zhang, Q., Sato, M., Chen, Y. Y., Chen, F., & Ohkuma, T. (2019). Augmenting knowledge tracing by considering forgetting behavior. In: *Proceedings of the World Wide Web Conference*. San Francisco: ACM, 3101–3107.
- Nakagawa, H., Iwasawa, Y., & Matsuo, Y. (2019). Graph-based knowledge tracing: Modeling student proficiency using graph neural network. In: *Proceedings of 2019 IEEE/WIC/ACM International Conference on Web Intelligence*. Thessaloniki: IEEE, 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.
- Pandey, S., & Karypis, G. (2019). A self attentive model for knowledge tracing. In: *Proceedings of the 12th International Conference on Education Data Mining*. Montreal: IEDMS, 1–6.
- Pavlik, P. I., Cen, H., & Koedinger, K. R. (2009). Performance factors analysis—A new alternative to knowledge tracing. In: *Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*. Brinton: IOS Press, 531–538.
- Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L., & Sohl-Dickstein, J. (2015). Deep knowledge tracing. In: *Proceedings of the 29th International Conference on Neural Information Processing Systems*. Montreal: MIT Press, 1–9.
- Ritter, S., Yudelson, M., Fancsali, S. E., & Berman, S. R. (2016). How mastery learning works at scale. In: *Proceedings of the*

- Third (2016) ACM Conference on Learning @ Scale*. Edinburgh: ACM, 71–79.
- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1), 61–80.
- Shen, S. H., Liu, Q., Chen, E. H., Wu, H., Huang, Z. Y., Zhao, W. H., Su, Y., Ma, H. P., & Wang, S. J. (2020). Convolutional knowledge tracing: Modeling individualization in student learning process. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Xi'an: ACM, 1857–1860.
- Sun, X. J., Liu, Q., Zhang, K., Shen, S. H., Yang, L. N., & Li, H. (2025a). Harnessing code domain insights: Enhancing programming knowledge tracing with large language models. *Knowledge-Based Systems*, 317, 113396.
- Sun, X. J., Zhang, K., Liu, Q., Shen, S. H., Wang, F., Guo, Y. X., & Chen, E. H. (2025b). DASKT: A dynamic affect simulation method for knowledge tracing. *IEEE Transactions on Knowledge and Data Engineering*, 37(4), 1714–1727.
- Tong, S. W., Liu, Q., Huang, W., Huang, Z. Y., Chen, E. H., Liu, C. R., Ma, H. P., & Wang, S. J. (2020). Structure-based knowledge tracing: An influence propagation view. In: *Proceedings of 2020 IEEE International Conference on Data Mining*. Sorrento: IEEE, 541–550.
- Vie, J. J., & Kashima, H. (2019). Knowledge tracing machines: Factorization machines for knowledge tracing. In: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*. Honolulu: AAAI, 750–757.
- Wei, Y. A., & Jiang, B. (2024). Interpretable cognitive state prediction via temporal fuzzy cognitive map. *IEEE Transactions on Learning Technologies*, 17, 514–526.
- Wei, Y., Zhou, Y. Z., Jiang, Y. H., & Jiang, B. (2024). Enhancing explainability of knowledge learning paths: Causal knowledge networks. In: *Proceedings of the Human-Centric Explainable AI in Education and the Leveraging Large Language Models for Next Generation Educational Technologies Workshops (HEXED-L3MNGET 2024) Co-located with 17th International Conference on Educational Data Mining*. Atlanta: CEUR-WS.org, 1–8.
- Wilson, K. H., Karklin, Y., Han, B. J., & Ekanadham, C. (2016). Back to the basics: Bayesian extensions of IRT outperform neural networks for proficiency estimation. In: *Proceedings of the 9th International Conference on Educational Data Mining*. Raleigh: IEDMS, 1–6.
- Yang, Y., Shen, J., Qu, Y. R., Liu, Y. F., Wang, K. R., Zhu, Y. M., Zhang, W. N., & Yu, Y. (2021). GIKT: A graph-based interaction model for knowledge tracing. In: *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*. Ghent: Springer, 299–315.
- Yudelson, M. V., Koedinger, K. R., & Gordon, G. J. (2013). Individualized Bayesian knowledge tracing models. In: *Proceedings of the 16th International Conference on Artificial Intelligence in Education*. Memphis: Springer, 171–180.
- Zhang, J. N., Shi, X. J., King, I., & Yeung, D. Y. (2017). Dynamic key-value memory networks for knowledge tracing. In: *Proceedings of the 26th International Conference on World Wide Web*. Perth: ACM, 765–774.
- Zhang, M. Y., Zhu, X. N., Zhang, C. H., Qian, W. C., Pan, F., & Zhao, H. (2023). Counterfactual monotonic knowledge tracing for assessing students' dynamic mastery of knowledge concepts. In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. Birmingham: ACM, 3236–3246.