HCGKT: Hierarchical Contrastive Graph Knowledge Tracing with Multi-level Feature Learning

Anonymous Author(s)

Paper ID:6670

Abstract. Knowledge Tracing (KT) aims to predict learners' future performance by modeling their historical interaction data. Despite recent advances in attention-based KT models, challenges persist in capturing hierarchical features between questions and knowledge components (KCs)[1], handling noisy data and modeling semantic relationships. We propose Hierarchical Contrastive Graph Knowledge Tracing (HCGKT), which combines hierarchical graph filtering attention, adversarial contrastive learning, and graph convolutional networks to address these challenges. Experiments on three datasets demonstrate our model's superior performance in both prediction accuracy and interpretability. We have provided all the datasets and code at `https://anonymous.4open.science/r/HCGKT-5DDC/`.

Keywords: knowledge tracing · hierarchical · contrastive learning · graph

1 Introduction

Knowledge tracing (KT) is a fundamental task in educational data mining that aims to estimate learners' knowledge states and predict their future performance by modeling historical interaction data [2]. With billions of daily learning interactions generated on global educational platforms, accurate knowledge tracing helps identify struggling learners and supports personalized learning path planning to improve learning outcomes [23].

With the rapid development of deep learning technology, the field of KT has made breakthrough progress. The introduction of sequence models, such as RNN [6], can effectively capture the dynamic evolution of learners' knowledge states, but faces challenges in dealing with long sequences, such as gradient vanishing and computational efficiency. In recent years, attention-based [30] KT models(such as SAINT [4], AKT [5], simpleKT [16]) have gradually become the mainstream solution in this field by establishing direct correlations within sequences. These models not only overcome the problem of long-range dependencies but also adaptively focus on key historical information [25].

[1] A knowledge component (KC) is a generalization of everyday terms like concept, principle, fact, or skill.

Despite the success of attention-based KT models, several limitations remain. First, students' knowledge mastery is hierarchical, progressing from surface to deep understanding through cognitive processing. However, existing models suffer from over-smoothing [32], where the attention distribution tends to be uniform, making it difficult to distinguish between different levels of knowledge features and learning abilities. Second, current methods inadequately handle noisy data from real-world scenarios, such as random answering behaviors (such as carelessness, guessing, or fatigue). Third, while learning effectiveness relies on connecting new knowledge with existing knowledge, current approaches focus too heavily on question and KC IDs, neglecting the semantic information of the questions and the association between KCs. Although there are attempts to improve modeling using graph neural networks, the joint modeling of semantic information and graph structure remains insufficient.

Therefore, in this paper, we propose HCGKT (Hierarchical Contrastive Graph Knowledge Tracing) to address the aforementioned challenges. Specifically, we use the hierarchical graph filtering attention mechanism to address over-smoothing by gradually extracting multi-level feature representations of students, capturing the hierarchical knowledge relationships between questions and KCs. In addition, we combine contrastive learning with adversarial perturbations to handle noisy student interaction data. By injecting random perturbations during training, we simulate complex educational data scenarios and leverage contrastive learning to extract robust features. Finally, we employ a Graph Convolutional Network [11] (GCN) to jointly model semantic information and structural relationships between questions and KCs. By introducing real-world KC semantics and relationship graphs, we effectively address the challenge of insufficient semantic and structural modeling, enhancing knowledge tracing capabilities. We conduct comprehensive experiments on three benchmark datasets (Algebra2005, Bridge2algebra2006 [27], XES3G5M [17]) and achieve superior performance in terms of AUC and Accuracy metrics.

The main contributions of this paper are:

- We address the over-smoothing problem by capturing hierarchical knowledge relationships between questions and KCs through multi-level feature representations.
- We enhance model robustness against noisy student interaction data by combining adversarial perturbations with contrastive learning.
- We improve knowledge tracing performance by jointly modeling semantic information and structural relationships in question-and-KC graph.

2 Related Work

2.1 Deep Learning Based KT Models

The emergence of deep learning has led to four main types of DLKT models [29,18,37]. Sequence-based models, pioneered by DKT, use LSTM networks to capture learning patterns and have been enhanced with features like question

difficulty and time intervals [24]. Attention-based models like AKT have been developed to better handle long-term dependencies in learning sequences [5]. Graph-based models, represented by GKT, focus on modeling relationships between KCs using graph neural networks [33]. Finally, memory-enhanced models explicitly capture the dynamics between KCs and student knowledge states through memory networks [1].

2.2 Robustness of KT Models

Recent studies have approached the robustness of KT models from multiple perspectives. SparseKT addresses the overfitting issue of attention-based KT models on small-scale datasets by introducing a k-selection module and sparsification strategies [8]. HD-KT designs dual anomaly detectors based on knowledge states and student profiles to effectively identify and handle anomalous interactions in the learning process [19]. Focusing on the quality of question representations, ABQR proposes an adversarial bootstrapped representation method that enhances model performance through multi-objective multi-round feature adversarial graph augmentation [28].

2.3 Relation Modeling for KT Models

Relation modeling has become essential in KT. KQN encodes learning activities into knowledge vectors and computes interactions via dot products [12], while DKVMN employs a dual-memory architecture to capture KC relationships and track knowledge mastery [38]. GKT reformulates the task using graph neural networks to model KC relationships explicitly.

3 The HCGKT Framework

3.1 Problem Definition

In online education with $|S|$ students and $|Q|$ questions, KT predicts students' future performance by analyzing their historical responses. For each learner, the system maintains a temporal interaction sequence $I = \{I_1, I_2, \ldots, I_t\}$. Each interaction record I_i is characterized by a quaternary tuple $I_i = (q_i, c_i, r_i, t_i)$, where $q_i \in Q$ denotes the question presented to the learner during the i-th interaction. $c_i = \{j | j \in N_{q_i}\}$ represents the set of KCs associated with question q_i, where N_{q_i} encompasses all KCs required by question q_i. $r_i \in \{0, 1\}$ indicates the learner's response, with 1 denoting correct and 0 denoting incorrect. t_i represents the temporal indicator, either as a timestamp or discrete time step. KT aims to predict a learner's performance on the next question q_{t+1} by computing the conditional probability: $P(r_{t+1} = 1 | I_1, I_2, \ldots, I_t, \{q_{t+1}, c_{t+1}\})$.

3.2 The Framework Overview

In this section, we present the framework of our HCGKT model (shown in Figure 1) that consists of five main components: (1) Semantic-Structure Fusion Module that integrates both semantic information and structural relationships between questions and KCs through parallel GCN [11] structures (See Section 3.1); (2) Robust Contrastive Learning Module that enhances model robustness using adversarial perturbations and contrastive learning (See Section 3.2); (3) Input Layer that processes and encodes the student interaction sequences (See Section 3.3); (4) Hierarchical Feature Distillation Module that leverages graph-enhanced order-aware attention mechanism to capture multi-level feature representations (See Section 3.4); and (5) KT Prediction Layer that uses a two-layer fully connected network to produce the final prediction output (See Section 3.5).

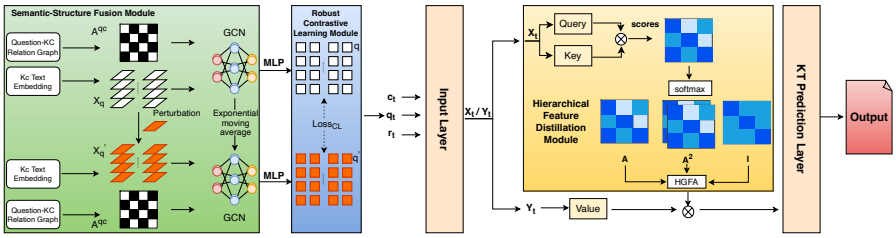

Fig. 1. The overview of the HCGKT framework.

3.3 Semantic-Structure Fusion Module

To effectively capture both semantic information and structural relationships between questions and KCs, we propose a semantic-structure fusion module. For each KC, we utilize the BAAI General Embeddings (BGE) model [3] to obtain its semantic representation. Given a KC's textual description t_i, we generate its semantic embedding $\mathbf{v}_i = \text{BGE}(t_i)$ where $\mathbf{v}_i \in \mathbb{R}^d$ represents the KCs semantic embedding. This embedding captures the semantic meaning of each KC through contextualized language representations.

Furthermore, we construct a directed graph to model the relationships between questions and KCs. For a dataset with N_q questions and N_c KCs, we define an adjacency matrix $\mathbf{M} \in \mathbb{R}^{N_q \times N_c}$, where $M_{i,j}$ equals 1 if question i contains KC j, and 0 otherwise. For each question q_i, we first aggregate the semantic embeddings of its associated KCs through mean pooling:

$$\mathbf{r}_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{v}_j$$

where $\mathbf{r}_i \in \mathbb{R}^d$, and $\mathcal{N}_i$ denotes the set of KCs connected to question i. We then employ a GCN to integrate the structural information:

$$\mathbf{Z} = \text{GCN}\left(\mathbf{R}, \mathbf{M}\right) = \text{ReLU}(\mathbf{MRW} + \mathbf{b})$$

where $\mathbf{R} \in \mathbb{R}^{N_q \times d}$ is the matrix of aggregated question representations, $\mathbf{W} \in \mathbb{R}^{d \times d}$ and $\mathbf{b} \in \mathbb{R}^d$ are learnable parameters, and ReLU is the rectified linear unit activation function. The final output $\mathbf{Z} \in \mathbb{R}^{N_q \times d}$ represents the enhanced question representations that incorporate both semantic and structural information.

3.4 Robust Contrastive Learning Module

To enhance the robustness of question representations against noisy student interaction data, we follow [28] and introduce contrastive learning with adversarial perturbations. To simulate potential noise in educational data and enhance model robustness, we generate adversarial perturbations $\boldsymbol{\delta} \in \mathbb{R}^d$ from a uniform distribution $\boldsymbol{\delta} \sim \mathcal{U}(-\varepsilon, \varepsilon)$, where ε controls the magnitude of the perturbation. This perturbation is designed to create challenging but meaningful variations in the input space.

For a question embedding $\mathbf{q} \in \mathbb{R}^d$ and its relationship matrix $\mathbf{M}$, we design two complementary representations, i.e. the original ($\mathbf{z}_p$) and perturbed ($\mathbf{z}_a$) representations, through different GCN encoders to capture robust features, defined as follows:

$$\mathbf{z}_p = \mathrm{GCN}_1(\mathbf{q}, \mathbf{M}) \in \mathbb{R}^d; \quad \mathbf{z}_a = \mathrm{GCN}_2(\mathbf{q} + \boldsymbol{\delta}, \mathbf{M}) \in \mathbb{R}^d$$

The contrastive learning objective aims to maximize the consistency between original and perturbed representations of the same question:

$$\mathcal{L}_{\text{contrast}} = \frac{1}{2}\big(\ell(p(\mathbf{z}_p), \mathbf{z}_a) + \ell(p(\mathbf{z}_a), \mathbf{z}_p)\big)$$

where $p(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ is a prediction MLP that projects the representations into a space where the contrastive loss is computed, and $\ell(\cdot, \cdot)$ is a similarity measure.

3.5 Input Layer

Following the interaction encoding scheme of SimpleKT [16] and extending it, we design our input layer to effectively model student learning trajectories. Instead of using question difficulty vectors, we leverage the robust question representations learned from our contrastive learning module. The interaction sequences are encoded as follows:

$$\mathbf{x}_t = \mathbf{k}_t \oplus \mathbf{z}_a \quad \mathbf{y}_t = \mathbf{k}_t \oplus \mathbf{s}_t \quad \mathbf{k}_t = \mathbf{W}_k \mathbf{e}_t^c \quad \mathbf{s}_t = \mathbf{W}_s \mathbf{e}_t^r$$

where $\mathbf{k}_t \in \mathbb{R}^d$ represents the latent embedding of KC c_t, and $\mathbf{s}_t \in \mathbb{R}^d$ denotes the response embedding at timestamp t. $\mathbf{e}_t^c \in \mathbb{R}^s$ and $\mathbf{e}_t^r \in \mathbb{R}^2$ are the original one-hot vectors of the corresponding KC and response (correct/incorrect) respectively. $\mathbf{W}_k \in \mathbb{R}^{d \times s}$ and $\mathbf{W}_s \in \mathbb{R}^{d \times 2}$ are learnable transformation matrices.

The interaction embedding $\mathbf{x}_t$ incorporates both the KC information and the question representation $\mathbf{z}_a$ obtained from our contrastive learning module, where $\oplus$ denotes element-wise addition. The final interaction representation $\mathbf{y}_t$

combines the KC embedding with the response information, enabling our model to capture the complete interaction context.

3.6 Hierarchical Feature Distillation Module

Inspired by the graph filtering self-attention mechanism [10], we design a hierarchical feature distillation module to capture multi-level feature representations in KT sequences. By incorporating graph-enhanced attention, our module effectively models both local and global dependencies in students' learning trajectories.

Given question representations $\mathbf{x}$ and interaction representations $\mathbf{y}$, we first project $\mathbf{x}$ into query and key spaces, and $\mathbf{y}$ into value space to compute the initial attention scores through multi-head attention, where the attention scores $\mathbf{A}$ are calculated using scaled dot-product attention with $\text{softmax}(\mathbf{Q}\mathbf{K}^T/\sqrt{d_k})$. Here $\mathbf{Q}, \mathbf{K}$ are linear projections of question representations $\mathbf{x}$, and $\mathbf{V}$ is projected from interaction representations $\mathbf{y}$. To capture higher-order interaction patterns, we propose a Hierarchical Graph Filtering Attention (HGFA) mechanism:

$$\mathbf{A}_K = \mathbf{A} + (K-1)(\mathbf{A}^2 - \mathbf{A}) \quad \mathbf{P} = \omega_0\mathbf{I} + \omega_1\mathbf{A} + \omega_K\mathbf{A}_K$$

where $\mathbf{A}^2$ models second-order relationships, $\mathbf{A}_K$ aggregates higher-order interactions where $K \geq 2$ is a hyperparameter, and $\omega_K, \omega_0, \omega_1$ are learnable parameters that control the contribution of different interaction orders. The identity matrix $\mathbf{I}$ helps preserve local information while learning global patterns.

The next hidden state is computed as: $\mathbf{h}_{t+1} = \text{FFN}(\mathbf{P}\mathbf{V})$, where $\mathbf{V}$ represents the value vectors and FFN is a two-layer feed-forward network with ReLU activation. This hierarchical feature processing enables our model to capture both fine-grained knowledge state transitions and long-term learning patterns.

3.7 KT Prediction Layer

To make the final prediction, we employ a two-layer fully connected network that takes the concatenation of the hidden state $\mathbf{h}_{t+1}$ and the input embedding $\mathbf{x}_{t+1}$ as input:

$$\hat{r}_{t+1} = \sigma\left(\phi\left(\mathbf{W}_2 \cdot \phi\left(\mathbf{W}_1 \cdot [\mathbf{h}_{t+1}; \mathbf{x}_{t+1}] + \mathbf{b}_1\right) + \mathbf{b}_2\right)\right)$$

where σ, ϕ denote Sigmoid and ReLU functions respectively, and $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2$ are trainable parameters.

3.8 Overall Training Objective

The overall training objective consists of two parts: the KT loss and the contrastive learning loss: $\mathcal{L} = \mathcal{L}_{\text{kt}} + \lambda\mathcal{L}_{\text{contrast}}$, where $\mathcal{L}_{\text{kt}}$ is the binary cross-entropy loss for KT:

$$\mathcal{L}_{\text{kt}} = -\sum_{t}(r_{t+1} \cdot \log \hat{r}_{t+1} + (1 - r_{t+1}) \cdot \log(1 - \hat{r}_{t+1}))$$

λ is a coefficient that controls the contribution of the contrastive loss $\mathcal{L}_{\text{contrast}}$ defined in Section 3.2.

4 Experiments

4.1 Datasets

To evaluate our model's performance, we conduct experiments on three widely-used public educational datasets.

- **Algebra 2005-2006 (AL2005)**[2]**:** AL2005 was collected from the KDD Cup 2010 Educational Data Mining Challenge, containing step-by-step response data from students aged 13-14 solving algebraic problems. Unique questions are defined by concatenating problem name with step name.
- **Bridge to Algebra 2006-2007 (BD2006)**[3]**:** BD2006, also from the KDD Cup 2010 Challenge, consists of mathematical problems from students' interactions with intelligent tutoring systems. Unique questions are formed by combining problem and step identifiers.
- **XES3G5M**[4]**:** XES3G5M is a large-scale dataset from a Chinese online mathematics learning platform. The dataset contains student-question interactions where each question is tagged with multiple KCs, enabling detailed learning trajectory tracking.

Table 1. Data statistics of three widely used datasets.

Dataset	AL2005	BD2006	XES3G5M
# of interactions	809,694	1,817,458	5,549,635
# of students	574	1,145	18,066
# of questions	210,710	129,263	7,652
# of KCs	112	948	865

4.2 Settings and Implementation Details

In our empirical study, we employed 5-fold cross-validation across all datasets using Adam optimizer for up to 200 epochs. Model hyperparameters were optimized using Bayesian techniques, with performance evaluated via AUC and Accuracy metrics. The architecture parameters were selected from ranges: embedding and hidden dimensions [64, 256], transformer blocks [1, 2, 4], attention heads [4, 8]. We explored learning rates [1e-3, 2e-3, 1e-4, 1e-5], dropout rates

[2] https://pslcdatashop.web.cmu.edu/KDDCup
[3] https://pslcdatashop.web.cmu.edu/KDDCup
[4] https://github.com/ai4ed/XES3G5M

[0.05-0.5], random seeds [42, 3407], hierarchy levels k [1-5], adversarial parameters (step size [1-5], gradient clip [5.0-20.0]), and momentum decay [0.9-0.99]. Experiments were conducted on NVIDIA RTX 4090 GPUs using PyTorch, with all methods optimized via Weights & Biases[5] for fair comparison.

4.3 Baselines

We evaluate HCGKT against 20 competitive KT baselines: (1) **DKT** is the first RNN-based model for Knowledge Tracing [24]; (2) **DKT+** enhances DKT by adding reconstruction and waviness regularization terms [35]; (3) **DKT-F** extends DKT by incorporating forgetting behaviors [20]; (4) **KQN** encodes activities into vectors and computes interactions via dot products [12]; (5) **DKVMN** uses static and dynamic memory for KC relationships and knowledge mastery [38]; (6) **ATKT** is an LSTM model with adversarial perturbations to mitigate overfitting [7]; (7) **GKT** handles KC relations as time-series node classification via GNN [21]; (8) **SAKT** uses self-attention to model KC and interaction relationships [22]; (9) **SAINT** is a Transformer model with exercise encoding and response prediction [4]; (10) **AKT** uses Rasch model and self-attention to refine embeddings and interactions [5]; (11) **SKVMN** combines recurrent modeling with memory networks for learning processes [1]; (12) **HAWKES** introduces Hawkes process for temporal cross-effects modeling [31]; (13) **DeepIRT** combines IRT with DKVMN for improved explainability [34]; (14) **LPKT** tracks knowledge by directly modeling learning process [26]; (15) **AT-DKT** enhances DKT with question tagging and prior knowledge prediction [15]; (16) **simpleKT** combines Rasch model with time-aware attention for interaction modeling [16]; (17) **FoLiBiKT** adds linear bias to AKT to model forgetting behavior [9]; (18) **DTransformer** uses TCA and attention for diagnosis, enhanced by contrastive learning [36]; (19) **extraKT** enhances extrapolation using encoders with efficient position embedding [14]; (20) **stableKT** learns from short sequences with stable generalization to long sequences [13].

4.4 Overall Performance

We evaluate HCGKT against 20 baseline models on three datasets using AUC and Accuracy metrics. Bold font represents the best results, and underlined font denotes the second-best results. '±' represents 5-fold cross validation experimental results. As shown in Table 2, we can draw the following conclusions: (1) Our proposed HCGKT model consistently achieves the best performance across all three datasets in terms of both AUC and Accuracy metrics. Specifically, HCGKT achieves improvements of [AUC: 0.8456, 0.8270, 0.8250] and [Accuracy: 0.8235, 0.8623, 0.8285] respectively, outperforming all baseline methods. (2) Compared to attention-based models like SimpleKT and AKT, HCGKT shows substantial improvements. This can be attributed to our hierarchical graph filtering attention mechanism, which effectively addresses the over-smoothing problem

[5] `https://wandb.ai/`

by capturing multi-level interaction patterns in KT. (3) When comparing with graph-based methods like GKT, HCGKT demonstrates notable advantages (approximately 2-3% increase in both metrics). Our model's superior performance highlights the effectiveness of integrating real-world KC semantics and relationship graphs, providing a more comprehensive understanding of question-KC relationships. (4) Regarding model robustness, HCGKT exhibits remarkable stability across datasets, as evidenced by the small standard deviations (±0.0022, ±0.0010, ±0.0017 for AUC). Compared to models like ATKT (±0.0023, ±0.0008, ±0.0004) or SAKT (±0.0063, ±0.0008, ±0.0008), these consistent variations demonstrate our adversarial training strategy effectively enhances model robustness against noisy student interaction data.

Table 2. Performance comparison of AUC and Accuracy. The best values are in bold and the second-best values are underlined.

Method	AUC			Accuracy		
	AL2005	**BD2006**	**XES3G5M**	**AlL2005**	**BD2006**	**XES3G5M**
DKT	0.8149±0.0011	0.8015±0.0008	0.7852±0.0006	0.8097±0.0005	0.8553±0.0002	0.8173±0.0002
DKT+	0.8156±0.0011	0.8020±0.0004	0.7861±0.0002	0.8097±0.0007	0.8553±0.0003	0.8178±0.0001
DKT-F	0.8147±0.0013	0.7985±0.0013	0.7940±0.0006	0.8090±0.0005	0.8536±0.0004	0.8209±0.0003
KQN	0.8027±0.0015	0.7936±0.0014	0.7793±0.0006	0.8025±0.0006	0.8532±0.0006	0.8152±0.0002
DKVMN	0.8054±0.0011	0.7983±0.0009	0.7792±0.0004	0.8027±0.0007	0.8545±0.0002	0.8155±0.0001
ATKT	0.7995±0.0023	0.7889±0.0008	0.7783±0.0004	0.7998±0.0019	0.8511±0.0004	0.8145±0.0002
GKT	0.8110±0.0009	0.8046±0.0008	0.7727±0.0006	0.8088±0.0008	0.8555±0.0002	0.8135±0.0004
SAKT	0.7880±0.0063	0.7740±0.0008	0.7693±0.0008	0.7954±0.0020	0.8461±0.0005	0.8124±0.0002
SAINT	0.7775±0.0017	0.7781±0.0013	0.8074±0.0007	0.7791±0.0016	0.8411±0.0065	0.8177±0.0006
AKT	0.8306±0.0019	0.8208±0.0007	0.8207±0.0008	0.8124±0.0011	0.8587±0.0005	0.8273±0.0007
SKVMN	0.7463±0.0022	0.7287±0.0052	0.7514±0.0005	0.7837±0.0023	0.8406±0.0005	0.8075±0.0003
HAWKES	0.8210±0.0012	0.8068±0.0010	0.7921±0.0007	0.8115±0.0009	0.8559±0.0005	0.8188±0.0003
DeepIRT	0.8040±0.0013	0.7976±0.0006	0.7785±0.0005	0.8037±0.0009	0.8543±0.0003	0.8150±0.0002
LPKT	0.8268±0.0004	0.8056±0.0008	0.8163±0.0002	0.8154±0.0008	0.8547±0.0005	0.8264±0.0001
AT-DKT	0.8246±0.0019	0.8104±0.0009	0.7932±0.0004	0.8144±0.0008	0.8560±0.0005	0.8198±0.0004
simpleKT	0.8254±0.0003	0.8160±0.0006	0.8163±0.0006	0.8083±0.0005	0.8579±0.0003	0.8246±0.0005
FoLiBiKT	0.8316±0.0012	0.8208±0.0016	0.8214±0.0007	0.8135±0.0016	0.8585±0.0008	0.8271±0.0006
DTransformer	0.8188±0.0025	0.8093±0.0009	0.8144±0.0006	0.8043±0.0021	0.8555±0.0007	0.8248±0.0004
extraKT	0.8317±0.0021	0.8110±0.0009	0.8200±0.0008	0.8110±0.0009	0.8605±0.0012	0.8263±0.0010
stableKT	0.8351±0.0008	0.8252±0.0003	0.8195±0.0012	0.8130±0.0007	0.8606±0.0002	0.8257±0.0003
HCGKT	**0.8456±0.0022**	**0.8270±0.0010**	**0.8250±0.0017**	**0.8235±0.0016**	**0.8623±0.0004**	**0.8285±0.0019**

4.5 Ablation Study

To evaluate the effectiveness of different components in HCGKT, we conduct comprehensive ablation studies by removing three key modules: Semantic-Structure Fusion Module (SFM), Robust Contrastive Learning Module (CL), and Hierarchical Graph Filtering Attention (HGFA). The experimental results in Table 3 reveal several important findings:

Impact of Individual Components (1) The CL module proves to be the most crucial component, with its removal leading to the largest performance degradation across all datasets (AUC decreases of 1.35%, 0.76%, and 0.46% on AL2005, BD2006, and XES3G5M respectively). This suggests that the adversarial perturbation-based robustness enhancement significantly contributes to model performance, particularly in handling noisy student interaction data. (2) The absence of HGFA results in consistent performance drops (AUC decreases

of 0.58%, 0.59%, and 0.36%), highlighting the importance of capturing hierarchical relationships in student interaction sequences. (3) Removing SFM also negatively impacts performance, though to a relatively smaller extent. This can be attributed to the fact that the number of KCs is substantially smaller than the number of questions in these datasets, which weakens the impact of KC semantic information and KC-question relationships.

Component Interaction Analysis (1) Removing both CL and HGFA results in a significant performance drop (AUC decreases of 1.40%, 0.89%, 0.55% on AL2005, BD2006, and XES3G5M respectively). The CL module provides robust and stable representations through adversarial perturbations and contrastive learning, while the HGFA captures complex, hierarchical dependencies among the problems. Together, they ensure the model has both noise-resistant features and a sophisticated understanding of relationships. When both are removed, the model loses its foundational robustness and the ability to capture higher-order relationships, leading to a substantial decrease in performance. (2) Removing both CL and SFM leads to a noticeable performance drop (AUC decreases of 0.81%, 0.76%, 0.58%). The CL module is crucial for ensuring noise resistance, while the SFM module integrates semantic and structural information between questions and KCs. Although SFM has a less direct impact than CL, their simultaneous removal diminishes both robustness and semantic fusion. The drop is significant, yet the model still retains some performance due to contributions from the remaining components. (3) Removing both HGFA and SFM causes a moderate performance decrease (AUC drops slightly but remains high). HGFA captures higher-order dependencies between problems, and SFM integrates the semantic and structural features of the Kcs. While both modules refine the model's understanding, the CL module's robust representation mitigates the impact of their removal. As a result, the performance decline is smaller compared to the other two experiments, with the core robustness from CL remaining intact.

4.6 Visualization Analysis

To validate our modules' effectiveness, we visualize and compare the enhanced question embeddings with randomly initialized embeddings. As shown in Figure 2(a), through integrating both semantic information of KCs and structural relationships between KCs and questions, questions with similar KCs form distinct clusters in the embedding space. This demonstrates that our approach successfully captures not only the semantic features of KCs but also effectively models the relationships between questions using KCs as bridges, naturally grouping questions that share the same KCs. In contrast, random embeddings (Figure 2(b)) show scattered distribution without meaningful structural relationships.

To showcase our HGFA mechanism's effectiveness, we compare attention score distributions before and after applying HGFA in Figure 3. The visualization shows that compared to the baseline model (Figure 3(a)), the attention distribution after HGFA processing (Figure 3(b)) exhibits stronger attention

Table 3. Average AUC and Accuracy component analysis on all datasets.

Models	AL2005	BD2006	XES3G5M
AUC			
-w/o CL	0.8321±0.0013	0.8194±0.0015	0.8204±0.0012
-w/o SFM	0.8404±0.0018	0.8209±0.0014	0.8230±0.0023
-w/o HGFA	0.8398±0.0025	0.8211±0.0017	0.8214±0.0020
-w/o CL+SFM	0.8375±0.0035	0.8194±0.0014	0.8192±0.0012
-w/o CL+HGFA	0.8316±0.0010	0.8181±0.0012	0.8195±0.0009
-w/o HGFA+SFM	0.8385±0.0009	0.8203±0.0016	0.8211±0.0012
HCGKT	**0.8456±0.0022**	**0.8270±0.0010**	**0.8250±0.0017**
Accuracy			
-w/o CL	0.8124±0.0015	0.8573±0.0014	0.8270±0.0007
-w/o SFM	0.8206±0.0011	0.8602±0.0007	0.8285±0.0010
-w/o HGFA	0.8188±0.0018	0.8595±0.0015	0.8264±0.0019
-w/o CL+SFM	0.8188±0.0027	0.8590±0.0018	0.8261±0.0006
-w/o CL+HGFA	0.8131±0.0017	0.8576±0.0013	0.8263±0.0005
-w/o HGFA+SFM	0.8185±0.0010	0.8595±0.0011	0.8273±0.0004
HCGKT	**0.8235±0.0016**	**0.8623±0.0004**	**0.8285±0.0019**

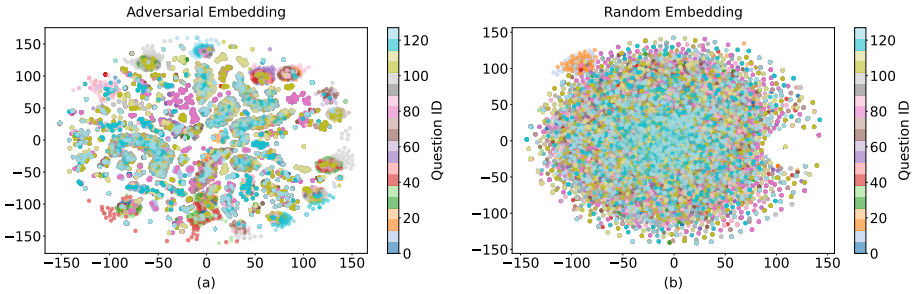

Fig. 2. Visualization of question embeddings: (a) Question embeddings enhanced by SFM and CL; (b) Randomly initialized question embeddings.

weights (deeper red coloring), indicating the model's enhanced ability to focus on key information. This enhanced attention distribution demonstrates that HGFA successfully addresses the over-smoothing issue, enabling the model to extract deeper feature representations. Through this approach, the model can effectively capture both fine-grained knowledge state transitions and broader learning trajectories, achieving effective modeling of hierarchical cognitive patterns.

5 Conclusion

In this paper, we propose HCGKT to address the fundamental challenges in KT. Specifically, our HCGKT model effectively captures hierarchical cognitive patterns through a novel hierarchical graph filtering attention mechanism, which mitigates the over-smoothing problem while preserving multi-level feature representations. Furthermore, we integrate contrastive learning with adversarial perturbations to enhance model robustness against noisy student interactions, and

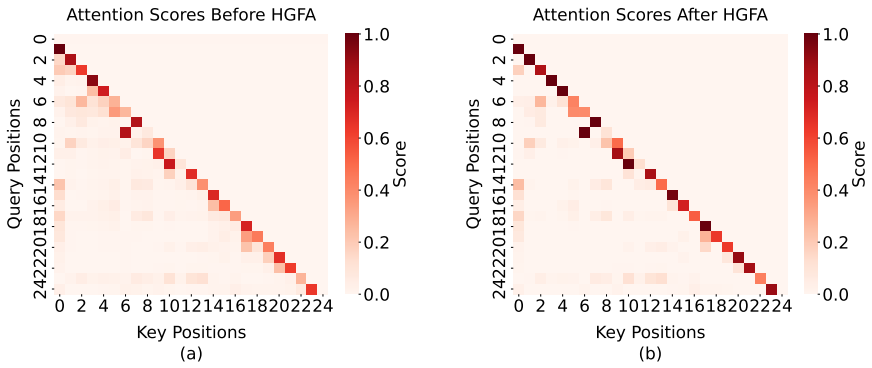

Fig. 3. Visualization of attention scores: (a) Attention scores before HGFA; (b) Attention scores after HGFA.

employ a GCN to jointly model semantic and structural relationships between questions and KCs. Experiments on real-world datasets show that HCGKT outperforms existing KT methods across multiple metrics.

References

1. Abdelrahman, G., Wang, Q.: Knowledge tracing with sequential key-value memory networks. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval. pp. 175–184 (2019)
2. Abdelrahman, G., Wang, Q., Nunes, B.: Knowledge tracing: A survey. ACM Computing Surveys **55**(11), 1–37 (2023)
3. Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., Liu, Z.: Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. arXiv preprint arXiv:2402.03216 (2024)
4. Choi, Y., Lee, Y., Cho, J., Baek, J., Kim, B., Cha, Y., Shin, D., Bae, C., Heo, J.: Towards an appropriate query, key, and value computation for knowledge tracing. In: Proceedings of the Seventh ACM Conference on Learning@ Scale. pp. 341–344 (2020)
5. Ghosh, A., Heffernan, N., Lan, A.S.: Context-aware attentive knowledge tracing. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2330–2339 (2020)
6. Graves, A., Graves, A.: Long short-term memory. Supervised sequence labelling with recurrent neural networks pp. 37–45 (2012)
7. Guo, X., Huang, Z., Gao, J., Shang, M., Shu, M., Sun, J.: Enhancing knowledge tracing via adversarial training. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 367–375 (2021)
8. Huang, S., Liu, Z., Zhao, X., Luo, W., Weng, J.: Towards robust knowledge tracing models via k-sparse attention. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2441–2445 (2023)

9. Im, Y., Choi, E., Kook, H., Lee, J.: Forgetting-aware linear bias for attentive knowledge tracing. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. pp. 3958–3962 (2023)
10. Kamalloo, E., Palm, R., Teichmann, M., Sayres, R.: Graph convolutions enrich the self-attention in transformers! arXiv preprint arXiv:2210.14310 (2022)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
12. Lee, J., Yeung, D.Y.: Knowledge query network for knowledge tracing: How knowledge interacts with skills. In: Proceedings of the 9th international conference on learning analytics & knowledge. pp. 491–500 (2019)
13. Li, X., Bai, Y., Guo, T., Liu, Z., Huang, Y., Zhao, X., Xia, F., Luo, W., Weng, J.: Enhancing length generalization for attention based knowledge tracing models with linear biases. In: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24). pp. 5918–5926 (2024)
14. Li, X., Bai, Y., Guo, T., Zheng, Y., Hou, M., Zhan, B., Huang, Y., Liu, Z., Gao, B., Luo, W.: Extending context window of attention based knowledge tracing models via length extrapolation. In: ECAI 2024, pp. 1479–1486. IOS Press (2024)
15. Liu, Z., Liu, Q., Chen, J., Huang, S., Gao, B., Luo, W., Weng, J.: Enhancing deep knowledge tracing with auxiliary tasks. In: Proceedings of the ACM Web Conference 2023. pp. 4178–4187 (2023)
16. Liu, Z., Liu, Q., Chen, J., Huang, S., Luo, W.: simplekt: a simple but tough-to-beat baseline for knowledge tracing. arXiv preprint arXiv:2302.06881 (2023)
17. Liu, Z., Liu, Q., Guo, T., Chen, J., Huang, S., Zhao, X., Tang, J., Luo, W., Weng, J.: Xes3g5m: A knowledge tracing benchmark dataset with auxiliary information. Advances in Neural Information Processing Systems **36** (2024)
18. Lu, Y., Wang, D., Meng, Q., Chen, P.: Towards interpretable deep learning models for knowledge tracing. In: Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21. pp. 185–190. Springer (2020)
19. Ma, H., Yang, Y., Qin, C., Yu, X., Yang, S., Zhang, X., Zhu, H.: Hd-kt: Advancing robust knowledge tracing via anomalous learning interaction detection. In: Proceedings of the ACM on Web Conference 2024. pp. 4479–4488 (2024)
20. Nagatani, K., Zhang, Q., Sato, M., Chen, Y.Y., Chen, F., Ohkuma, T.: Augmenting knowledge tracing by considering forgetting behavior. In: The world wide web conference. pp. 3101–3107 (2019)
21. Nakagawa, H., Iwasawa, Y., Matsuo, Y.: Graph-based knowledge tracing: modeling student proficiency using graph neural network. In: IEEE/WIC/ACM International Conference on Web Intelligence. pp. 156–163 (2019)
22. Pandey, S., Karypis, G.: A self-attentive model for knowledge tracing. arXiv preprint arXiv:1907.06837 (2019)
23. Paudel, P.: Online education: Benefits, challenges and strategies during and after covid-19 in higher education. International Journal on Studies in Education (IJonSE) **3**(2) (2021)
24. Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L.J., Sohl-Dickstein, J.: Deep knowledge tracing. Advances in neural information processing systems **28** (2015)
25. Pu, S., Becker, L.: Self-attention in knowledge tracing: Why it works. In: International Conference on Artificial Intelligence in Education. pp. 731–736. Springer (2022)

26. Shen, S., Liu, Q., Chen, E., Huang, Z., Huang, W., Yin, Y., Su, Y., Wang, S.: Learning process-consistent knowledge tracing. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. pp. 1452–1460 (2021)
27. Stamper, J., Koedinger, K., Pavlik Jr, P., Baker, R., Rossi, L.: A temporal analysis of student learning trajectories using kdd cup 2010 data. Journal of Educational Data Mining **2**(1), 107–128 (2010)
28. Sun, J., Yu, F., Liu, S., Luo, Y., Liang, R., Shen, X.: Adversarial bootstrapped question representation learning for knowledge tracing (2023)
29. Valero-Leal, E., Carlon, M.K.J., Cross, J.S.: A shap-inspired method for computing interaction contribution in deep knowledge tracing. In: International conference on artificial intelligence in education. pp. 460–465. Springer (2023)
30. Vaswani, A.: Attention is all you need. Advances in Neural Information Processing Systems (2017)
31. Wang, C., Ma, W., Zhang, M., Lv, C., Wan, F., Lin, H., Tang, T., Liu, Y., Ma, S.: Temporal cross-effects in knowledge tracing. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining. pp. 517–525 (2021)
32. Wu, X., Ajorlou, A., Wu, Z., Jadbabaie, A.: Demystifying oversmoothing in attention-based graph neural networks. Advances in Neural Information Processing Systems **36** (2024)
33. Yang, Y., Liu, Z., Zhi, M., Liu, H., Huang, K., Hu, N., Zhang, Y.: Graph-based knowledge tracing: modeling student proficiency using graph neural network. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1869–1877 (2020)
34. Yeung, C.K.: Deep-irt: Make deep learning based knowledge tracing explainable using item response theory. arXiv preprint arXiv:1904.11738 (2019)
35. Yeung, C.K., Yeung, D.Y.: Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In: Proceedings of the fifth annual ACM conference on learning at scale. pp. 1–10 (2018)
36. Yin, Y., Dai, L., Huang, Z., Shen, S., Wang, F., Liu, Q., Chen, E., Li, X.: Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer. In: Proceedings of the ACM Web Conference 2023. pp. 855–864 (2023)
37. Zhan, B., Guo, T., Li, X., Hou, M., Liang, Q., Gao, B., Luo, W., Liu, Z.: Knowledge tracing as language processing: A large-scale autoregressive paradigm. In: International Conference on Artificial Intelligence in Education. pp. 177–191. Springer (2024)
38. Zhang, J., Shi, X., King, I., Yeung, D.Y.: Dynamic key-value memory networks for knowledge tracing. In: Proceedings of the 26th international conference on World Wide Web. pp. 765–774 (2017)