

SLAM: SEQUENTIAL LEARNING SIGNAL MODELING FOR MULTI-CONCEPT KNOWLEDGE TRACING

Rui Jia^{1,2,3*}, Yuan-Hao Jiang^{2,3,4*}, Yaomin Shen⁵, Cheng Chen¹,
Yuan Liu¹, Zi-Wei Chen⁶, Yongquan Dong^{1†}

¹School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, China

²Shanghai Institute of Artificial Intelligence for Education, East China Normal University, SH, China

³Lab of Artificial Intelligence for Education, East China Normal University, Shanghai, China

⁴Shanghai Jiao Tong University, Shanghai, China

⁵XR System Application Research Center, Nanchang Research Institute, Zhejiang University, China

⁶School of Information Engineering, Jingdezhen Ceramic University, Jingdezhen, China

ABSTRACT

Knowledge Tracing (KT) aims to align with learners' evolving knowledge states by modeling sequential learning signals, thereby enabling educators to anticipate potential dropout risks. However, most existing KT approaches assume each question involves only one concept, known as Single Concept KT (SCKT). In real-world educational scenarios, a question often involve multiple concepts, while the absence of Multi Concept KT (MCKT) methods forces models to capture only isolated factors such as forgetting, guessing, or difficulty. Moreover, compressing inherently multi-concept learning signals into single-concept representations results in irreversible information loss, which we describe as falling into Hume's Limits of Imagination Paradox, the inability of an agent to imagine what has never been experienced. This distortion leads to overfitting and restricts both accuracy and interpretability. To address these issues, we propose Sequential Learning signal Modeling (SLAM) for MCKT. SLAM introduces relation-aware embeddings to represent multi-concept questions, employs a monotonic attention mechanism to model forgetting behaviors in sequential signals, and designs decoders to jointly capture concept mastery, guessing, slipping, and difficulty. Experiments on two large-scale datasets show that SLAM outperforms state of the art models by up to 4% in AUC, confirming its effectiveness and interpretability for MCKT.

Index Terms— Knowledge Tracing, Learning Signal Modeling, AI for Education, Intelligent Tutoring System

1. INTRODUCTION

In educational data mining, a knowledge component (KC) is regarded as the minimal unit of knowledge required to solve an item [1, 2]. Knowledge Tracing (KT) dynamically aligns learners' mastery of KCs with sequential learning signals, which not only supports understanding of individual learning processes but also provides practical value for preventing dropout risks [3, 4]. KT research dates back to the rise of Intelligent Tutoring Systems (ITSs) and Massive Open Online Courses (MOOCs) [5, 6], where massive amounts of learner interaction data became available.

Over the past three decades, researchers have investigated multiple factors underlying learning processes, such as forgetting effects

*These authors contributed equally to this work.

†Corresponding author.

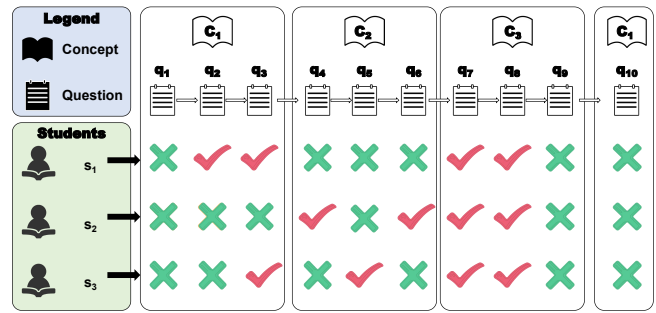


Fig. 1: Sequential learning signals with multiple factors.

[1, 7], concept mastery [8, 9, 10], guessing and slipping probabilities [11, 12], item difficulty [13, 14], and learning styles [15]. These factors provide multiple perspectives for understanding learning signals and help facilitate personalized learning [16, 17, 18].

With the success of deep learning, *Deep Knowledge Tracing (DKT)* [19] first introduced *Long Short-Term Memory (LSTM)* networks [20] to model sequential learning data. This was followed by the development of the *Dynamic Key-Value Memory Network (DKVMN)* [21], the *Self-Attentive Knowledge Tracing (SAKT)* model [22], the *Attentive Knowledge Tracing (AKT)* framework [22], the *Sequential Self-Attentive Knowledge Tracing (SSAKT)* model [23], and the *Forgetting-Aware Linear Bias Knowledge Tracing (Folibikt)* approach [24], which significantly improved predictive performance. However, these methods predominantly fall under *Single-Concept Knowledge Tracing (SCKT)*, assuming each item involves only one KC. In real educational settings, items frequently involve multiple KCs. Forcing inherently multi-concept signals into single-concept representations causes irreversible information loss. We describe this as **Hume's Limits of Imagination Paradox** [25]: just as humans cannot imagine what they have never experienced, KT models constrained to single-concept inputs inevitably suffer from representational distortion, overfitting, and reduced interpretability.

Figure 1 presents a simplified learning scenario with three learners, three KCs, and ten items. Interactions between learners, items, and associated KCs form sequential learning signals. The figure highlights several key factors commonly observed in learning: (i) differences in learning style; (ii) forgetting effects; (iii) variation in

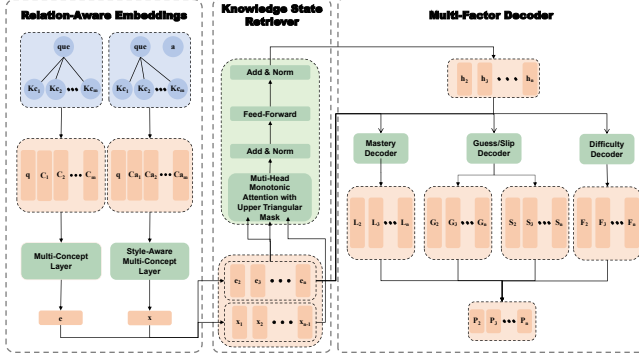


Fig. 2: SLAM framework for multi-concept knowledge tracing.

concept mastery across learners; (iv) guessing and slipping; and (v) item difficulty. This example underscores the need for KT models to integrate heterogeneous signals to achieve accurate and interpretable *Multi-Concept Knowledge Tracing (MCKT)*. Yet existing approaches typically address only subsets of these factors and lack a unified framework for multi-concept signal modeling.

To address these limitations, we propose **Sequential Learning signal Modeling (SLAM)** for MCKT. SLAM introduces relation-aware embeddings to represent multi-concept items, employs a monotonic attention mechanism to capture forgetting behaviors, and incorporates decoders to jointly model mastery, guessing, slipping, and difficulty. The main contributions of this work are:

1. We propose *relation-aware embeddings* to distinguish items involving multiple KCs, going beyond prior approaches that average concept vectors.
2. We design a unified framework that integrates multiple learning factors—learning style, forgetting, mastery, guessing/slipping, and difficulty—into sequential signal modeling.
3. We evaluate SLAM on two large-scale real-world datasets, showing up to 4% AUC improvement over SOTA KT models, while enhancing interpretability of the learning process.

2. METHOD

The proposed SLAM consists of three core modules: (i) relation-aware multi-concept embeddings, (ii) a knowledge-level retriever, and (iii) a multi-factor decoder. Figure 2 illustrates the SLAM architecture and its modular components. The input to SLAM is a sequence of learner–item interactions, and the output is the probability of correctly answering the next item. At time step n , the historical sequence is denoted as $X_n = \{x_1, x_2, \dots, x_{n-1}\}$, where $x_i = (q_i, Kc_i, a_i)$ represents an interaction at step i . Here, q_i is the item identifier, $Kc_i = \{Kc_{i1}, Kc_{i2}, \dots, Kc_{im}\}$ is the set of knowledge components (KCs) associated with the item, m denotes the variable number of KCs per item, and $a_i \in \{0, 1\}$ is the learner’s response (0 = incorrect, 1 = correct). The model estimates the probability P_n that the learner answers item q_n correctly.

SLAM processes these signals in three stages. First, relation-aware multi-concept embeddings represent items and interactions. Second, a knowledge-level retriever with monotonic attention extracts hidden knowledge states from past signals while modeling forgetting effects. Finally, the multi-factor decoder separates four interpretable factors: concept mastery, guessing probability, slipping

Algorithm 1 Relation-aware Question Embedding

Require: Item identifier que and its concept set $K_c = \{K_1, K_2, \dots, K_m\}$
Ensure: Relation-aware question embedding e

- 1: Initialize parameters $W_Q^C, W_K^C, W_V^C, W_{que}$
- 2: $q = que \cdot W_{que}, C = K_c \cdot W_K^C$
- 3: $Query^C = C \cdot W_Q^C, Value^C = C \cdot W_V^C$
- 4: $Key^C = [q]_m \cdot W_K^C$
- 5: $w_i^C = \frac{\exp(\langle Query_i^C, Key_i^C \rangle)}{\sum_{j=1}^m \exp(\langle Query_j^C, Key_j^C \rangle)}$
- 6: $e = \sum_{i=1}^m w_i^C \cdot Value_i^C$
- 7: **return** e

probability, and item difficulty, to predict learner responses. The following subsections describe each module in detail.

2.1. Relation-Aware Multi-Concept Embeddings

To distinguish different items involving the same concept set and to address the over-parameterization of simple averaging methods, we design relation-aware embeddings. This approach jointly incorporates the item identifier, its associated concept set, and the learner’s response outcome to generate a question embedding e and an interaction embedding x .

Before presenting the pseudocode, we define the notations: que denotes the item identifier, $K_c = \{K_1, \dots, K_m\}$ represents the set of concepts associated with the item, $a \in \{0, 1\}$ is the learner’s response (0 for incorrect, 1 for correct). The embedded representations are denoted as q, C , and CL ; $Query, Key$, and $Value$ denote mapped vectors; w_i is the attention weight; e is the relation-aware question embedding; and x is the relation-aware interaction embedding enriched with learning style. The embedding procedures are described in the following algorithms. Algorithm 1 presents the construction of relation-aware embeddings for questions. Algorithm 2 extends this to learner–item interactions by incorporating response outcomes and learning style information. The embeddings e and x produced here are then passed into the knowledge-level retriever described in Section 2.2, ensuring smooth integration of relation-aware representations into the subsequent learning process.

2.2. Knowledge-Level Retriever

The knowledge-level retriever is based on a Transformer encoder with two task-specific modifications: (i) a monotonic attention mechanism that incorporates time intervals and exponential decay to simulate forgetting, and (ii) an upper-triangular mask to prevent information leakage from future interactions.

The inputs are the question embedding sequence $E = \{e_2, \dots, e_n\}$ and the interaction embedding sequence $X = \{x_1, \dots, x_{n-1}\}$. These are projected into query, key, and value spaces:

$$E^Q = EW^Q, \quad E^K = EW^K, \quad X^V = XW^V \quad (1)$$

The monotonic attention weights are adjusted by a time distance $d(n, i)$ and an exponential decay factor:

$$\alpha_{n,i} = \text{Softmax} \left(\text{Mask} \left(\frac{\exp(-\theta \cdot d(n, i)) (E_n^Q)^T E_i^K}{\sqrt{\text{dim}_k}} \right) \right) \quad (2)$$

Multi-head attention aggregates these weights with X^V , and the outputs are processed by a feed-forward network with residual and normalization layers. This yields the hidden knowledge states $\{h_2, \dots, h_n\}$.

Algorithm 2 Relation-aware Interaction Embedding

Require: Interaction (que, K_c, a) , where que is the item id, $K_c = \{K_1, K_2, \dots, K_m\}$ is the concept set, and a is the response

Ensure: Relation-aware interaction embedding x

- 1: Initialize parameters $W_Q^L, W_K^L, W_V^L, W_{que}$
 - 2: $q = que \cdot W_{que}$
 - 3: **for** $j = 1$ to m **do**
 - 4: $C_{a_j} = K_{c_j} + |K| \cdot a$
 - 5: **end for**
 - 6: $CL = LSTM(Ca)$
 - 7: $Query^L = CL \cdot W_Q^L, Value^L = CL \cdot W_V^L$
 - 8: $Key^L = [q]_m \cdot W_K^L$
 - 9: $w_i^L = \frac{\exp(\langle Query_i^L, Key_i^L \rangle)}{\sum_{j=1}^m \exp(\langle Query_j^L, Key_j^L \rangle)}$
 - 10: $x = \sum_{i=1}^m w_i^L \cdot Value_i^L$
 - 11: **return** x
-

2.3. Multi-Factor Decoder

The multi-factor decoder transforms hidden states $\{h_i\}$ into four interpretable factors via linear and sigmoid layers: concept mastery L_i , guessing G_i , slipping S_i , and difficulty F_i . First, difficulty adjusts mastery:

$$L_i^f = L_i \times (1 - F_i) \quad (3)$$

Then the probability of a correct response is computed as:

$$P_i = \text{sigmoid}\left(L_i^f(1 - S_i) + (1 - L_i^f)G_i\right) \quad (4)$$

This design ensures that predictions are both accurate and interpretable through the lens of multiple educational factors.

3. EXPERIMENT

3.1. Experimental Setup and Datasets

We evaluate SLAM on two real-world datasets and compare it with KT baselines.

Datasets. We use two widely adopted datasets: ASSISTment [26] and EdNet [27]. ASSISTment contains about 8k learners, 110 concepts, nearly 7k questions, and 1.2M interactions, making it a standard KT benchmark. EdNet is the largest public dataset with 715k learners, 188 concepts, 12k questions, and 173M interactions. Notably, nearly half of EdNet’s questions involve 2–7 concepts, while most in ASSISTment involve only a single concept, offering complementary evaluation settings.

Baseline Methods and Evaluation Metrics. We compare SLAM against several representative KT methods, including Deep Knowledge Tracing (DKT), Dynamic Key-Value Memory Network (DKVMN), Self-Attentive Knowledge Tracing (SAKT), Attentive Knowledge Tracing (AKT), and Sequential Self-Attentive Knowledge Tracing (SSAKT) [28]. Effectiveness is measured by the area under the ROC curve (AUC) [29] and accuracy (ACC), which capture the ability to predict binary-valued learner responses.

Training and Testing. SLAM is implemented in PyTorch. We adopt 5-fold cross-validation, splitting learners into 60% training, 30% validation, and 10% testing. The validation set is used for early stopping and parameter tuning. To ensure efficiency, sequences longer than 200 are truncated or segmented, and each question is limited to at most 7 concepts (padded if fewer). Parameter initialization follows prior work [30], and most models complete one epoch in under 10 seconds, with a maximum of 300 epochs.

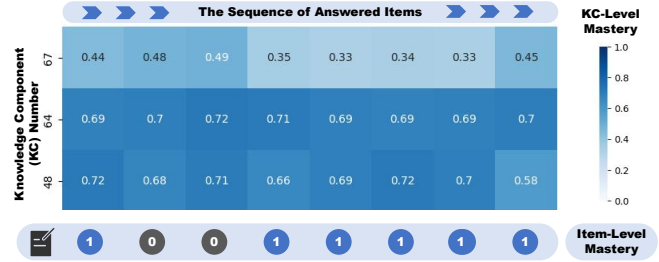


Fig. 3: KC-level mastery and item-level outcomes of a learner across the answer sequence. The vertical axis represents KCs, the horizontal axis denotes the sequence of answered items, and the color scale indicates the predicted mastery values. Groundtruth correctness (0/1) is shown at the bottom as item-level outcomes.

Table 1: Performance (%) of KT methods on ASSISTment and EdNet datasets. Best results are **bold and underlined**.

Model	ASSISTment		EdNet	
	AUC	ACC	AUC	ACC
DKT-R	78.22	71.51	66.29	67.88
DKVMN-R	75.70	69.51	66.06	67.04
SAKT-R	77.28	71.01	71.22	69.34
AKT-R	79.26	72.22	74.96	71.42
SSAKT-R	80.24	71.11	74.51	67.51
Folibikt-R	80.92	73.57	75.71	71.81
SLAM (Ours)	81.71	74.04	76.76	72.58

3.2. Performance Prediction

Table 1 reports the performance (%) of all Knowledge Tracing (KT) methods across the two datasets in predicting future learner responses. We present averages over five test folds. For reference, “X-R” indicates that the embedding module of model X is replaced with the relation-aware multi-concept embeddings proposed in this work. The best results are highlighted in **bold and underlined**.

We observe that SLAM outperforms all baselines on both ASSISTment and EdNet. Compared with the sub-optimal SSAKT, SLAM improves AUC by 1.83% on ASSISTment and 3.01% on EdNet, confirming the benefit of integrating multiple learning factors for KT. Furthermore, attention-based models (SAKT, AKT, SSAKT, Folibikt, and SLAM) consistently surpass memory network-based DKVMN and recurrent network-based DKT, highlighting the flexibility and effectiveness of attention mechanisms in capturing rich sequential information.

3.3. Ablation Study

To evaluate the core innovations of SLAM—relation-aware multi-concept embeddings and multi-factor decoding—we conducted two types of ablation experiments, and the results are summarized in Table 2. First, for embeddings, we replaced baseline models’ original embeddings with relation-aware embeddings (-R) or average embeddings (-M). Results show consistent improvements for -R over -M. For example, on ASSISTment, AUC gains are 2.1% for DKT and 3.0% for SAKT, confirming the effectiveness of relation-aware embeddings in modeling multi-concept items.

Second, we removed specific factors (learning style, forgetting, guessing and slipping, and difficulty) from SLAM while keeping

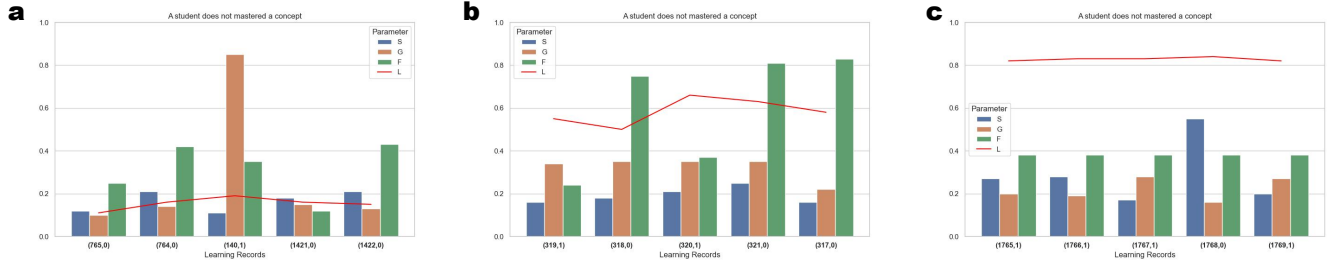


Fig. 4: Interpretability analysis of three typical learning trajectories: (a) low mastery with guessing effects; (b) partial mastery with difficulty effects; (c) high mastery with occasional slipping.

Table 2: Ablation results (%) on ASSISTment and EdNet. Rows above the horizontal line are baselines with alternative embeddings; rows below are SLAM (Ours) and its factor-removal variants. Underlined numbers indicate the better result between -M and -R for a same model. **Bold numbers** indicate the overall best results. These results demonstrate both the effectiveness of the relation-aware embedding (-R) and the synergy of the SLAM components.

Model	ASSISTment		EdNet	
	AUC	ACC	AUC	ACC
DKT-M	76.16	70.22	65.11	66.43
DKT-R	<u>78.22</u>	<u>71.51</u>	<u>66.29</u>	<u>67.88</u>
DKVMN-M	75.56	69.48	65.88	66.91
DKVMN-R	<u>75.70</u>	<u>69.51</u>	<u>66.06</u>	<u>67.04</u>
SAKT-M	74.32	70.14	70.01	68.54
SAKT-R	<u>77.28</u>	<u>71.01</u>	<u>71.22</u>	<u>69.34</u>
AKT-M	78.66	71.95	74.34	70.93
AKT-R	<u>79.26</u>	<u>72.22</u>	<u>74.96</u>	<u>71.42</u>
SLAM-L	81.14	73.61	76.10	72.01
SLAM-Forget	80.92	73.23	75.96	72.06
SLAM-SG	81.48	73.92	76.14	72.14
SLAM-Diff	81.41	73.80	76.39	72.35
SLAM (Ours)	81.71	74.04	76.76	72.58

other settings unchanged. Results demonstrate that forgetting behavior is the most critical factor: removing it decreases AUC by nearly 1% on ASSISTment and 1.0% on EdNet. In contrast, difficulty contributes the least. Overall, SLAM achieves the best performance across both datasets, validating the proposed design.

3.4. Interpretability Analysis

In knowledge tracing, beyond accurate performance estimation, interpretability is essential for explaining why learners succeed or fail. We study the interpretability of SLAM from two perspectives. First, we visualize how learners' mastery of KCs evolves along their answer sequence. As shown in Figure 3, SLAM captures the dynamics of KC-level mastery: after a correct answer, the mastery of related KCs increases, while after an incorrect answer, the mastery decreases. This visualization demonstrates that SLAM can align its outputs with realistic learning trajectories.

Second, we analyze the educational significance of SLAM's parameters through typical learning cases. Figure 4 illustrates three representative learner trajectories. In case (a), a learner who has not

mastered a concept answered five questions with four incorrect, and the mastery parameter L remained in a low range. However, one correct answer was still observed due to the guessing factor G . In case (b), a learner with partial mastery achieved two correct answers but failed four questions, where higher values of difficulty parameter F explained the errors despite moderate mastery ($L \in [0.4, 0.7]$). In case (c), a learner with high mastery ($L \in [0.7, 0.8]$) answered four of five correctly. The occasional error was attributed to the slipping parameter S , reflecting real-world situations where mistakes occur even when concepts are well mastered.

In summary, SLAM not only predicts learner performance but also provides interpretable outputs reflecting real educational contexts. By linking item outcomes with KC mastery and factors (L, G, S, F), SLAM reveals mechanisms behind learner responses and offers insights into diverse behaviors.

4. CONCLUSION

We proposed SLAM, a deep knowledge tracing model that integrates relation-aware multi-concept embeddings with multiple learning factors, including learning style, forgetting, mastery, guessing, slipping, and difficulty. SLAM enhances both prediction accuracy and interpretability by modeling how these factors interact within the learning process. Experiments on two real-world datasets demonstrate consistent improvements over existing baselines. In future work, we plan to combine SLAM with multidimensional item response theory to further explore learner behavior and optimize both performance and interpretability.

5. ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China, under the Grants 61872168 and 62577036, and the Special Foundation for Interdisciplinary Talent Training in 'AI Empowered Psychology / Education' of East China Normal University, under Grant 2024JCRC-03. This work was also jointly supported in part by the Fundamental Research Funds for the Central Universities, and the ECNU Academic Innovation Promotion Program for Excellent Doctoral Students (both under Grant YBNLTS2025-008). This work was initiated while Rui Jia was at Jiangsu Normal University. Yuan-Hao Jiang contributed to this work while participating in an inter-university joint student program at Shanghai Jiao Tong University.

6. REFERENCES

- [1] Yuying Chen et al., “Tracking knowledge proficiency of students with educational priors,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 989–998.
- [2] Yuan-Hao Jiang et al., “MAS-KCL: Knowledge component graph structure learning with large language model-based agentic workflow,” *The Visual Computer*, vol. 41, no. 2025, pp. 6453–6464, May 2025.
- [3] Ghodai Abdelrahman et al., “Knowledge tracing: A survey,” *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–37, 2023.
- [4] Yuan-Hao Jiang et al., “Explainable learning outcomes prediction: Information fusion based on grades time-series and student behaviors,” in *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, Apr. 2025, CHI EA ’25, pp. 1–11, Association for Computing Machinery.
- [5] Chris Piech et al., “Learning program embeddings to propagate feedback on student code,” in *International conference on machine Learning*. PMLR, 2015, pp. 1093–1102.
- [6] Shuanghong Shen et al., “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*, 2020, pp. 1857–1860.
- [7] Yu Su et al., “Time-and-concept enhanced deep multidimensional item response theory for interpretable knowledge tracing,” *Knowledge-Based Systems*, vol. 218, pp. 106819, 2021.
- [8] Richard Scruggs et al., “Using knowledge tracing to predict students’ performance in cognitive training and math,” in *International Conference on Artificial Intelligence in Education*. Springer, 2022, pp. 410–413.
- [9] Deliang Wang et al., “Traditional knowledge tracing models for clustered students,” *The Educational Review, USA*, vol. 4, no. 12, pp. 244–251, 2020.
- [10] Francisco Cervantes-Pérez et al., “Bayesian knowledge tracing for navigation through marzano’s taxonomy,” *International Journal of Interactive Multimedia and Artificial Intelligence*, 2021.
- [11] Wenbin Gan et al., “Modeling learner’s dynamic knowledge construction procedure and cognitive item difficulty for knowledge tracing,” *Applied Intelligence*, vol. 50, pp. 3894–3912, 2020.
- [12] Sein Minn et al., “Deep knowledge tracing and dynamic student classification for knowledge tracing,” in *IEEE International Conference on Data Mining*. IEEE, 2018, pp. 1182–1187.
- [13] Harold Pashler et al., “Predicting the optimal spacing of study: A multiscale context model of memory,” *Advances in neural information processing systems*, vol. 22, 2009.
- [14] Jianwen Sun et al., “Ensemble knowledge tracing: Modeling interactions in learning process,” *Expert Systems with Applications*, vol. 207, pp. 117680, 2022.
- [15] Hoa-Huy Nguyen et al., “A model to create a personalized online course based on the student’s learning styles,” *Education and Information Technologies*, vol. 29, no. 1, pp. 571–593, 2024.
- [16] Yuan-Hao Jiang et al., “Agentic workflow for education: Concepts and applications,” in *Proceedings of the 33rd International Conference on Computers in Education*, Chennai, India, Dec. 2025, Asia-Pacific Society for Computers in Education.
- [17] Ruijia Li et al., “How real is AI tutoring? Comparing simulated and human dialogues in one-on-one instruction,” in *Proceedings of the 33rd International Conference on Computers in Education*, Chennai, India, Dec. 2025, Asia-Pacific Society for Computers in Education.
- [18] Yuan-Hao Jiang et al., “Synchronizing Verbal Responses and Board Writing for Multimodal Math Instruction with LLMs,” in *NeurIPS’24: Conference and Workshop on Neural Information Processing Systems, the 4th Workshop on Mathematical Reasoning and AI*, Vancouver, Canada, Dec. 2024, pp. 46–59, Neural Information Processing Systems Foundation.
- [19] Chris Piech et al., “Deep knowledge tracing,” *Advances in neural information processing systems*, vol. 28, 2015.
- [20] Alex Graves and Alex Graves, “Long short-term memory,” *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.
- [21] Jiani Zhang et al., “Dynamic key-value memory networks for knowledge tracing,” in *Proceedings of the 26th international conference on World Wide Web*, 2017, pp. 765–774.
- [22] Aritra Ghosh et al., “Context-aware attentive knowledge tracing,” in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 2330–2339.
- [23] Xuelong Zhang et al., “Sequential self-attentive model for knowledge tracing,” in *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia*. Springer, 2021, pp. 318–330.
- [24] Yoonjin Im et al., “Forgetting-aware linear bias for attentive knowledge tracing,” in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 3958–3962.
- [25] David Hume, *An Enquiry Concerning Human Understanding*, A. Millar, London, 1748, Reprinted by Hackett Publishing, 1993. Edited by Eric Steinberg.
- [26] Thanaporn Patikorn et al., “Assistments longitudinal data mining competition special issue: a preface,” *Journal of Educational Data Mining*, vol. 12, no. 2, pp. i–xi, 2020.
- [27] Youngduck Choi et al., “Ednet: A large-scale hierarchical dataset in education,” in *Artificial Intelligence in Education: 21st International Conference*. Springer, 2020, pp. 69–73.
- [28] Xuelong Zhang et al., “Sequential self-attentive model for knowledge tracing,” in *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks*. Springer, 2021, pp. 318–330.
- [29] Yan Zhuang et al., “A robust computerized adaptive testing approach in educational question retrieval,” in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 416–426.
- [30] Xavier Glorot et al., “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings*, 2010, pp. 249–256.