Revisiting Applicable and Comprehensive Knowledge Tracing in Large-Scale Data

Yiyun Zhou[1], Wenkang Han[1], and Jingyuan Chen[1] (✉)

Zhejiang University {yiyunzhou, wenkangh, jingyuanchen}@zju.edu.cn

A Appendix

Due to space limitations, the main text cannot include all details. Here, we have supplemented the details mentioned in the main text, including:

– Summary of DLKT Models from 2015-2025 in Terms of Applicability and Comprehensiveness (A.1)
– Detailed Introduction to LSTM (A.2)
– Detailed Introduction to xLSTM (A.3)
– Dataset Description and Processing Methods (A.4)
– Baseline Description (A.5)
– Additional Experimental Results (A.6)

A.1 Summary of DLKT Models from 2015-2025 in Terms of Applicability and Comprehensiveness

Table 5 summarizes the DLKT models in terms of applicability and comprehensiveness in top AI/ML conferences/journals from 2015-2025.

A.2 Detailed Introduction to LSTM

Long Short-Term Memory (LSTM) [31] overcomes the short-term memory limitations of Recurrent Neural Networks (RNN) [18] caused by the vanishing gradient[3] [28, 29] by introducing cell state and gating mechanisms into the network. Fig. 6 shows the architecture of LSTM at time step t. The core concepts of LSTM include cell state and various gate structures. The cell state acts as a pathway for transmitting relevant information, allowing information to be passed along the sequence chain, which can be viewed as the network's memory. Theoretically, during sequence processing, the cell state can continuously carry relevant information. Thus, information obtained at earlier time steps can be transmitted to cells at later time steps, which helps mitigate the impact of short-term memory.

[3] The vanishing gradient refers to the phenomenon where, during model training, as time step increases, the gradient is continuously multiplied by the weight matrix during backpropagation, potentially causing it to shrink rapidly towards zero, resulting in very slow weight updates in the network.

Conference/Journal	Model	Applicability	Comprehensiveness
AAAI	**KTM** [67]	✓	✗
	IKT [49]	✓	✗
	QIKT [9]	✓	✓
	DAKTN [70]	✓	✓
CIKM	**MF-DAKT** [78]	✓	✗
	LFBKT [10]	✓	✗
	RKT [54]	✗	✗
	FoLiBiKT [35]	✗	✗
	CPKT [69]	✓	✗
	SFKT [79]	✓	✗
	CMKT [80]	✓	✗
	Sinkt [19]	✓	✗
	LOKT [25]	✓	✗
COLING	**KVFKT** [23]	✓	✗
ICDM	**DKT-DSC** [50]	✓	✓
	SKT [66]	✓	✗
	CAKT [72]	✓	✗
ICLR	**simpleKT** [44]	✗	✗
	PSI-KT [83]	✓	✗
IJCAI	**stableKT** [38]	✗	✗
KDD	**AKT** [21]	✗	✗
	LPKT [60]	✗	✗
	LBKT [71]	✗	✗
	DyGKT [11]	✓	✗
	GRKT [16]	✓	✗
MM	**ATKT** [24]	✓	✓
	ABQR [63]	✓	✗
	PSKT [34]	✓	✓
	ReKT [62]	✓	✗
NIPS	**DKT** [55]	✓	✓
PKDD	**GIKT** [73]	✓	✗
	GMKT [81]	✓	✗
	CCKT [82]	✓	✗
SIGIR	**SKVMN** [1]	✗	✗
	HGKT [65]	✓	✗
	CKT [61]	✓	✗
	IEKT [45]	✗	✗
	DIMKT [59]	✓	✗
	sparseKT [33]	✗	✗
TKDE	**EKT** [41]	✓	✗
	DGMN [2]	✓	✓
	LPKT-S [58]	✗	✗
	XKT [32]	✓	✗
TOIS	**MRT-KT** [15]	✗	✗
	DGEKT [14]	✓	✓
	MAN [27]	✗	✗
	FDKT [39]	✓	✗
	ELAKT [56]	✓	✓
WSDM	**HawkesKT** [68]	✗	✗
	AdaptKT [12]	✓	✗
	CoKT [46]	✓	✗
WWW	**DKVMN** [77]	✗	✗
	DKT-F [51]	✓	✓
	CL4KT [37]	✗	✗
	DTransformer [76]	✗	✗
	AT-DKT [43]	✓	✓
	MIKT [64]	✗	✗
	QDCKT [40]	✓	✗
	HD-KT [47]	✓	✗
	DisKT [84]	✗	✗

Table 5. Summary of the applicability and comprehensiveness of DLKT models in top AI/ML conferences/journals from 2015-2025. ✓ and ✗ indicate strong and weak applicability and comprehensiveness, respectively. The gray background indicates that the code is not open-source, and its applicability and comprehensiveness are inferred from the method section of the paper.

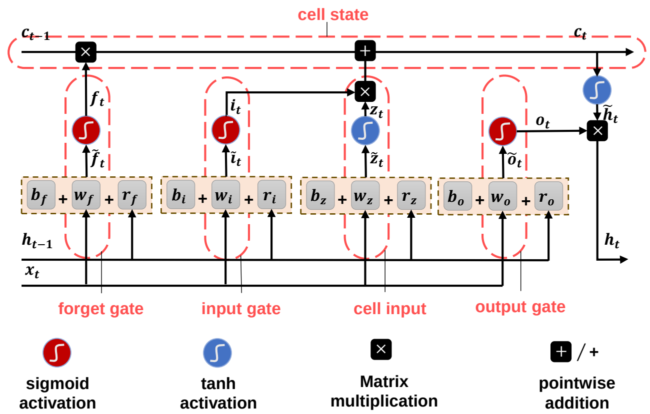

Fig. 6. Architecture of LSTM.

Additionally, LSTM addresses the short-term memory issue of RNNs by introducing internal gating mechanisms (*i.e.*, forget gate [20], input gate, and output gate) to regulate information flow. Specifically, LSTM uses the Tanh activation function (with output values always in the range (-1, 1)) to help regulate the neural network output and employs the Sigmoid activation function in its gate structures. The Sigmoid function is similar to the Tanh function, but its output range is (0, 1), which aids in updating or forgetting data, as any number multiplied by 0 becomes 0 (this information is forgotten), and any number multiplied by 1 remains unchanged (this information is fully preserved). This allows the network to understand which data is unimportant and should be forgotten, and which data is important and should be preserved. The cell state update rule (*i.e.*, the constant error carousel [30]) for LSTM at time step t is:

$$\begin{aligned}
f_t &= \sigma(\tilde{f}_t), \quad \tilde{f}_t = w_f^\top x_t + r_f\, h_{t-1} + b_f, \\
i_t &= \sigma(\tilde{i}_t), \quad \tilde{i}_t = w_i^\top x_t + r_i\, h_{t-1} + b_i, \\
z_t &= \varphi(\tilde{z}_t), \quad \tilde{z}_t = w_z^\top x_t + r_z\, h_{t-1} + b_z, \\
c_t &= f_t\, c_{t-1} + i_t\, z_t, \\
o_t &= \sigma(\tilde{o}_t), \quad \tilde{o}_t = w_o^\top x_t + r_o\, h_{t-1} + b_o, \\
h_t &= o_t\, \tilde{h}_t, \quad \tilde{h}_t = \varphi(c_t),
\end{aligned} \tag{10}$$

where the weight vectors w_f, w_i, w_z, and w_o correspond to the input weights between the input x_t and the forget gate, input gate, cell input, and output gate, respectively. The weights r_f, r_i, r_z, and r_o correspond to the recurrent weights between the hidden state h_{t-1} and the forget gate, input gate, cell input, and output gate, respectively. b_f, b_i, b_z, and b_o are the corresponding bias terms. $\varphi(\cdot)$ is the activation function for the cell input or hidden state (*e.g.*, Tanh), and $\sigma(\cdot)$ is the Sigmoid activation function, *i.e.*, $\sigma(x) = \frac{1}{1+exp(-x)}$.

In summary, the forget gate in LSTM determines which relevant information from previous time steps should be preserved, the input gate decides which important information from the current input should be added, and the output gate determines the next hidden state. Previous work [22] has shown that each gate structure is crucial. Recently, LSTM has been revisited and greatly improved, with the revised LSTM known as xLSTM [7]. xLSTM enhances the traditional LSTM structure, aiming to improve LSTM's performance and scalability with large-scale data. Subsequently, a series of studies on xLSTM have been applied to various fields such as computer vision [4, 85, 17] and time series [3].

A.3 Detailed Introduction to xLSTM

Stabilized Long Short-Term Memory To enable LSTM to revise storage decisions, sLSTM introduces an exponential activation function along with normalizer state and stabilization. Unlike the Sigmoid activation function (*i.e.*, S-shaped function) mentioned in Appendix A.2, where it becomes very challenging for the model to decide what to forget or retain as input values get higher, sLSTM uses an exponential function instead, providing a broader output range, indicating that sLSTM can better revise storage decisions. However, after introducing the exponential function, output values tend to surge as input values increase and do not naturally normalize outputs between 0 and 1 as the Sigmoid function does. Therefore, sLSTM introduces normalizer state, which is a function of the forget gate and input gate, to normalize the hidden state. The update rule for the sLSTM cell state at time step t is:

$$
\begin{aligned}
f_t &= \sigma(\tilde{f}_t)\,\mathrm{OR}\,\exp(\tilde{f}_t),\\
\tilde{f}_t &= w_f^\top\, x_t + r_f\, h_{t-1} + b_f,\\
i_t &= \exp(\tilde{i}_t), \quad \tilde{i}_t = w_i^\top\, x_t + r_i\, h_{t-1} + b_i,\\
z_t &= \varphi(\tilde{z}_t), \quad \tilde{z}_t = w_z^\top\, x_t + r_z\, h_{t-1} + b_z,\\
c_t &= f_t\, c_{t-1} + i_t\, z_t,\\
n_t &= f_t\, n_{t-1} + i_t,\\
o_t &= \sigma(\tilde{o}_t), \quad \tilde{o}_t = w_o^\top\, x_t + r_o\, h_{t-1} + b_o,\\
h_t &= o_t\, \tilde{h}_t, \quad \tilde{h}_t = c_t / n_t,
\end{aligned}
\tag{11}
$$

where the weight vectors w_f, w_i, w_z, and w_o correspond to the input weights between the input x_t and the forget gate, input gate, cell input, and output gate, respectively. The weights r_f, r_i, r_z, and r_o correspond to the recurrent weights between the hidden state h_{t-1} and the forget gate, input gate, cell input, and output gate, respectively. b_f, b_i, b_z, and b_o are the corresponding bias terms. φ is the activation function for the cell input or hidden state (*e.g.*, Tanh), σ is the Sigmoid activation function, and exp is the exponential activation function.

Moreover, since the exponential activation function can easily cause overflow for large values, to prevent the exponential function from disrupting the forget gate and input gate, sLSTM uses an additional state m_t [48], which appears in logarithmic form, to counteract the effect of the exponential function and introduce stability:

$$\begin{aligned}
m_t &= \max(\log(f_t) + m_{t-1}, \log(i_t)),\\
i'_t &= \exp(\log(i_t) - m_t) = \exp(\tilde{i}_t - m_t),\\
f'_t &= \exp(\log(f_t) + m_{t-1} - m_t),
\end{aligned} \tag{12}$$

Matrix Long Short-Term Memory To enhance LSTM's memory ability to capture more complex data relationships and patterns, mLSTM introduces a matrix $C \in \mathbb{R}^{d\times d}$ to replace the scalar cell state $c \in \mathbb{R}$. Additionally, since LSTM is designed to process sequential data, which means it needs to process the output of the previous input in the sequence to handle the current input, this hinders parallelization and is the main culprit leading to the Transformer era. Therefore, mLSTM abandons this design concept. Specifically, mLSTM adopts the setting of Bidirectional Associative Memories (BAMs) [36, 5]: at time step t, mLSTM stores a pair of vectors, key $k_t \in \mathbb{R}^d$ and value $v_t \in \mathbb{R}^d$. At time step $t+\tau$, the value v_t is retrieved through a query vector $q_{t+\tau} \in \mathbb{R}^d$. mLSTM uses a covariance update rule ($C_t = C_{t-1} + v_t\, k_t^\top$) to store the key-value pair. The covariance update rule is equivalent to the Fast Weight Programmer [57]. Later, a new variant has emerged [6]: a constant decay rate multiplied by C_{t-1} and a constant learning rate multiplied by $v_t\, k_t^\top$. Similarly, in mLSTM, the forget gate corresponds to the decay rate, while the input gate corresponds to the learning rate. Furthermore, since the dot product between the query input and the normalizer state may approach zero, mLSTM uses the absolute value of the dot product and sets a lower bound to a threshold (*e.g.*, 1). The cell state update rule for mLSTM is:

$$\begin{aligned}
f_t &= \sigma(\tilde{f}_t)\,\mathrm{OR}\exp(\tilde{f}_t), \quad \tilde{f}_t = w_f^\top\, x_t + b_f,\\
i_t &= \exp(\tilde{i}_t), \quad \tilde{i}_t = w_i^\top\, x_t + b_i,\\
k_t &= \frac{1}{\sqrt{d}} W_k\, x_t + b_k,\\
v_t &= W_v\, x_t + b_v,\\
q_t &= W_q\, x_t + b_q,\\
C_t &= f_t\, C_{t-1} + i_t\, v_t\, k_t^\top,\\
n_t &= f_t\, n_{t-1} + i_t\, k_t,\\
o_t &= \sigma(\tilde{o}_t), \quad \tilde{o}_t = W_o^\top\, x_t + b_o,\\
h_t &= o_t \odot \tilde{h}_t, \quad \tilde{h}_t = C_t\, q_t / \max\{\left|n_t^\top\, q_t\right|, 1\},
\end{aligned} \tag{13}$$

Similarly, to stabilize the exponential function in mLSTM, mLSTM employs the same stabilization technique as sLSTM (see Eq. 12). The design of mLSTM supports highly parallelized processing, which not only improves computational efficiency but also allows the model to scale better to large datasets.

In addition, xLSTM introduces residual networks [26] to stack sLSTM or mLSTM, enabling xLSTM to effectively process complex sequential data while improving the training stability of the model in deep networks.

A.4 Dataset Description and Processing Methods

We provide a detailed description of the datasets used in our experiments and the methods employed for processing them.

We conduct extensive experiments on three of the latest large-scale benchmark datasets from different platforms: (i) Assist17[4] is the latest subset of the ASSISTments dataset released by Worcester Polytechnic Institute. ASSISTments is an online tutoring system that provides mathematics instruction and access services for students, widely used in mathematics courses for 4th to 12th-grade students in the United States. A key feature of ASSISTments is providing students with immediate feedback, allowing them to know whether their answers are correct after responding to questions. (ii) EdNet[5] is a substantial educational dataset collected by Santa, a multi-platform artificial intelligence tutoring service. Collected over two years, this dataset encompasses a wide range of student-system interactions across Android, iOS, and web platforms in Korea. It contains over 130 million learning interactions from approximately 780,000 students, making it one of the largest publicly available interactive education system datasets. The dataset is notable for its scale and hierarchical structure, offering rich insights into student activities and learning patterns. To ensure computational efficiency, we randomly selected 20,000 students from EdNet, similar to previous studies [42, 37, 13]. (iii) Comp[6], which is part of PTADisc, is specifically selected for KT tasks in computational thinking courses. PTADisc originates from PTA, an online programming teaching assistant system developed by PTA Educational Technology Co., Ltd. for universities and society, based on students. PTADisc is currently the largest dataset in the field of personalized learning, which also includes different courses of varying data scales, providing options for various types of learning.

Following the data preprocessing method in CL4KT [37], we exclude students with fewer than five interactions and all interactions involving unnamed concepts. Since a single question may involve multiple concepts, we convert the unique concept combinations within a single question into a new concept. The statistics after processing are shown in Table 6.

Datasets	**#students**	**#questions**	**#concepts**	**#interactions**
Assist17	1,708	3,162	411	934,638
EdNet	20,000	12,215	1,781	2,709,132
Comp	45,180	8,392	472	6,072,632

Table 6. Statistics of three datasets after processing.

[4] https://sites.google.com/view/assistmentsdatamining/dataset?authuser=0
[5] https://github.com/riiid/ednet
[6] https://github.com/wahr0411/PTADisc

Dataset	Step	5			10			15			20		
	Metric	AUC↑	ACC↑	RMSE↓	AUC	ACC	RMSE	AUC	ACC	RMSE	AUC	ACC	RMSE
EdNet	DKT	0.6767	0.6406	0.4705	0.6724	0.6363	0.4721	0.6688	0.6333	0.4731	0.6669	0.6322	0.4738
	SAKT	0.6737	0.6389	0.4718	0.6706	0.6370	0.4729	0.6661	0.6334	0.4741	0.6662	0.6328	0.4741
	AKT	0.6793	0.6388	0.4703	0.6750	0.6366	0.4717	0.6719	0.6333	0.4728	0.6699	0.6316	0.4735
	Mamba4KT	0.6655	0.6350	0.4753	0.6632	0.6312	0.4753	0.6609	0.6288	0.4763	0.6586	0.6273	0.4769
	DKVMN	0.6709	0.6366	0.4722	0.6682	0.6341	0.4731	0.6641	0.6319	0.4742	0.6626	0.6308	0.4747
	ATKT	0.6704	0.6371	0.4735	0.6669	0.6355	0.4747	0.6633	0.6323	0.4758	0.6625	0.6305	0.4762
	CL4KT	-	-	-	-	-	-	-	-	-	-	-	-
	Deep-IRT	0.6573	0.6250	0.4792	0.6546	0.6216	0.4808	0.6499	0.6185	0.4820	0.6467	0.6170	0.4816
	AT-DKT	0.6816	0.6442	0.4693	0.6787	0.6414	0.4703	0.6752	0.6375	0.4716	0.6722	0.6355	**0.4727**
	DKT2	**0.6853**	**0.6451**	**0.4683**	**0.6809**	**0.6420**	**0.4697**	**0.6771**	**0.6379**	**0.4709**	**0.6731**	**0.6360**	0.4732
Comp	DKT	0.7419	0.8097	0.3722	0.7303	0.8086	0.3745	0.7208	0.8085	0.3759	0.7128	0.8089	0.3765
	SAKT	0.7418	0.8098	0.3725	0.7307	0.8087	0.3746	0.7213	0.8086	0.3760	0.7130	**0.8092**	0.3766
	AKT	0.7384	0.8081	0.3737	0.7262	0.8069	0.3762	0.7213	0.8086	0.3759	0.7073	0.8077	0.3782
	Mamba4KT	0.7424	0.8097	0.3723	0.7310	0.8084	0.3746	**0.7225**	0.8085	0.3757	0.7137	0.8090	0.3765
	DKVMN	0.7397	0.8091	0.3729	0.7286	0.8084	0.3750	0.7201	0.8082	0.3762	0.7117	0.8089	0.3768
	ATKT	0.7405	0.8094	0.3726	0.7302	0.8086	0.3746	0.7192	0.8082	0.3763	0.7103	0.8087	0.3771
	CL4KT	0.7364	0.8070	0.3746	**0.7339**	0.8082	**0.3743**	0.7142	0.8066	0.3778	**0.7184**	0.8082	**0.3763**
	Deep-IRT	0.7372	0.8084	0.3737	0.7255	0.8075	0.3760	0.7159	0.8073	0.3773	0.7072	0.8079	0.3780
	AT-DKT	0.7440	0.8098	**0.3718**	0.7311	0.8086	0.3744	0.7212	0.8083	0.3758	0.7123	0.8091	0.3767
	DKT2	**0.7459**	**0.8103**	0.3722	0.7328	**0.8089**	0.3746	0.7219	**0.8093**	**0.3753**	0.7152	0.8090	0.3765

Table 7. Multi-step prediction performance of DKT2 and several representative baselines on EdNet and Comp.

A.5 Baseline Description

Here is a detailed description of the 18 baselines from 8 different categories in our experiment.

- **Deep sequential models**
  - **DKT** [55]: DKT is a pioneering model that utilizes Recurrent Neural Networks (RNNs), specifically a single-layer Long Short-Term Memory (LSTM) network, to directly model students' learning processes and predict their performance.
  - **DKT+** [75]: DKT+ is an enhanced version of DKT. It addresses the reconstruction and prediction inconsistency issues present in the DKT by introducing additional regularization terms to the loss function.
  - **DKT-F** [51]: DKT-F improves upon DKT by incorporating students' forgetting behaviors into the modeling process.
- **Attention-based models**
  - **SAKT** [53]: SAKT leverages self-attention networks to analyze and understand the complex relationships between concepts and a student's historical interactions with learning materials.
  - **AKT** [21]: AKT is an advanced KT model that incorporates a Rasch model to regularize concept and question embeddings and a modified Transformer architecture with adaptive attention weights computed by a distance-aware exponential decay to account for the time distance between questions and students' previous interactions.

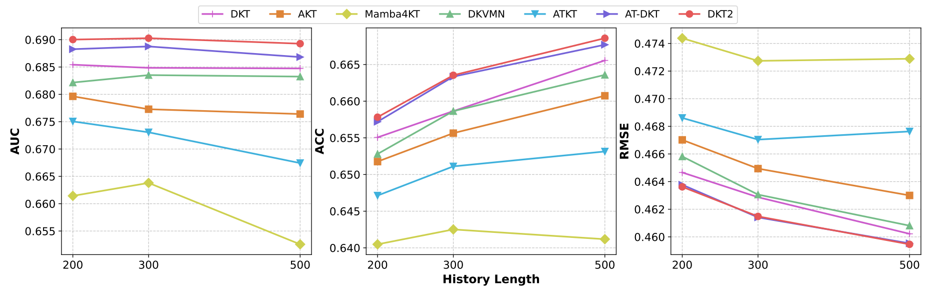

Fig. 7. The prediction performance of DKT2 and several representative baselines on EdNet with different history lengths.

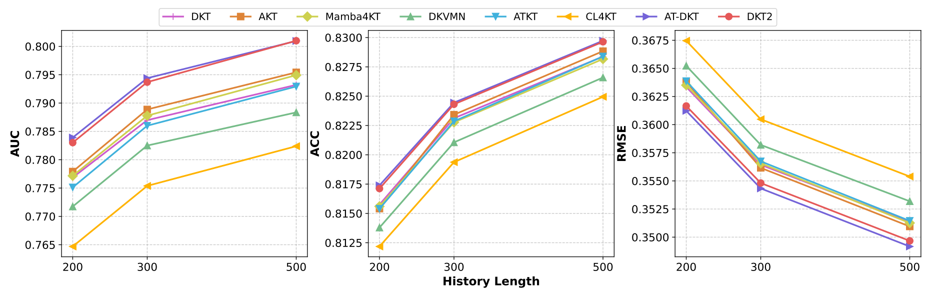

Fig. 8. The prediction performance of DKT2 and several representative baselines on Comp with different history lengths.

- **simpleKT** [44]: simpleKT is a simple but tough-to-beat baseline to KT that combines simplicity with robust performance.
- **FoLiBiKT** [35]: FoLiBi enhances attention-based KT models by incorporating a forgetting-aware linear bias mechanism. We introduce FoLiBi with AKT, namely FoLiBiKT.
- **sparseKT** [33]: sparseKT employs a k-selection module with soft-thresholding sparse attention (sparseKT-soft) and top-K sparse attention (sparseKT-topK) to focus on high-attention items, ensuring efficient and focused attention on the most relevant items.
- **DTransformer** [76]: DTransformer integrates question-level mastery with knowledge-level diagnosis through the use of Temporal and Cumulative Attention (TCA) and multi-head attention mechanisms. Additionally, a contrastive learning-based algorithm is used for enhancing the stability of the knowledge state diagnosis process.
- **stableKT** [38]: stableKT excels in length generalization, delivering stable and consistent performance across both short and long student interaction sequences. It employs a multi-head aggregation module that integrates dot-product and hyperbolic attention to capture hierarchical relationships between questions and their associated concepts.

– **Mamba-based models**

  - **Mamba4KT** [8]: By leveraging Mamba, a state-space model supporting parallelized training and linear-time inference, Mamba4KT achieves efficient resource utilization, balancing time and space consumption.
- **Graph-based models**
  - **GKT** [52]: GKT revolutionizes the traditional KT task by employing Graph Neural Networks (GNNs) to represent the relationships between concepts as a graph.
- **Memory-augmented models**
  - **DKVMN** [77]: DKVMN employs a static key matrix to capture the interrelationships among latent concepts and a dynamic value matrix for continuously updating and predicting a student's knowledge mastery in real-time.
  - **SKVMN** [1]: SKVMN combines recurrent modeling of DKT with memory networks of DKVMN to enhance tracking of learners' knowledge states over time.
- **Adversarial-based models**
  - **ATKT** [24]: ATKT is an attention-based LSTM model that employs adversarial training techniques to enhance generalization and reduce overfitting by applying perturbations to student interaction sequences.
- **Contrastive learning-based models**
  - **CL4KT** [37]: CL4KT employs contrastive learning on augmented learning histories to enhance representation learning by distinguishing between similar and dissimilar student learning patterns.
- **Other representative models**
  - **Deep-IRT** [74]: Deep-IRT is an explainable KT model that combines the DKVMN with Item Response Theory (IRT) to provide detailed insights into learner trajectories and concept difficulties, bridging deep learning capabilities with psychometric interpretability.
  - **AT-DKT** [43]: AT-DKT enhances the original DKT by incorporating two auxiliary learning tasks: one focused on predicting question tags and the other on evaluating individualized prior knowledge.

A.6 Additional Experimental Results

Due to space limitations in the main text, we have supplemented some additional experimental results here, including:

- Multi-step prediction results on EdNet and Comp datasets;
- Prediction results with varying history lengths on EdNet and Comp;
- Prediction results under three different input settings on EdNet and Comp;
- Ablation study on ACC and RMSE.

Multi-step Prediction Results Table 7 shows the multi-step (step=5, 10, 15, 20) prediction performance of DKT2 and several representative baselines from different categories on EdNet and Comp.

Datasets	Settings	Metrics	AKT	simpleKT	FoLiBiKT	sparseKT	DTransformer	stableKT	DKVMN	CL4KT	Deep-IRT
EdNet	▵	AUC↑	0.6083	0.6218	0.6098	0.6210	0.6140	0.6212	0.6195	-	0.6190
		ACC↑	0.5883	0.5938	0.5886	0.5916	0.5882	0.5907	0.5916	-	0.5896
		RMSE↓	0.4900	0.4896	0.4906	0.4901	0.4960	0.4894	0.4886	-	0.4892
	∘	AUC↑	0.6806	0.6903	0.6794	0.6853	0.6717	0.6775	0.6564	-	0.6577
		ACC↑	0.6309	0.6388	0.6325	0.6341	0.6258	0.6286	0.6138	-	0.6149
		RMSE↓	0.4798	0.4774	0.4815	0.4804	0.4895	0.4838	0.4815	-	0.4815
	•	AUC↑	0.6770	0.6832	0.6761	0.6813	0.6705	0.6801	0.6559	-	0.6559
		ACC↑	0.6303	0.6324	0.6291	0.6336	0.6245	0.6291	0.6141	-	0.6140
		RMSE↓	0.4824	0.4804	0.4879	0.4795	0.4925	0.4849	0.4818	-	0.4820
Comp	▵	AUC↑	0.7223	0.7208	0.7224	0.7171	0.7198	0.7194	0.7170	0.7184	0.7157
		ACC↑	0.7887	0.7873	0.7877	0.7857	0.7806	0.7866	0.7850	0.7852	0.7842
		RMSE↓	0.3919	0.3923	0.3921	0.3933	0.3955	0.3934	0.3938	0.3938	0.3946
	∘	AUC↑	0.8252	0.8251	0.8246	0.8217	0.8192	0.8160	0.7544	0.7465	0.7523
		ACC↑	0.8146	0.8145	0.8157	0.8165	0.8165	0.8115	0.7932	0.7920	0.7922
		RMSE↓	0.3612	0.3612	0.3607	0.3612	0.3621	0.3644	0.3837	0.3863	0.3845
	•	AUC↑	0.8160	0.8206	0.8157	0.8153	0.8107	0.8243	0.7450	0.7592	0.7430
		ACC↑	0.8097	0.8149	0.8160	0.8146	0.8118	0.8139	0.7907	0.7933	0.7897
		RMSE↓	0.3664	0.3617	0.3621	0.3632	0.3662	0.3616	0.3866	0.3830	0.3871

Table 8. The prediction performance of KT models with weak applicability and comprehensiveness in the last 5 steps on EdNet and Comp under three different input settings. The ▵ setting represents masking all interaction information (including questions, concepts and responses) for the last 5 steps, the ∘ setting represents masking the responses for the last 5 steps, without masking questions and concepts, and the • setting represents no masking, *i.e.*, predicting the responses under the regular setting.

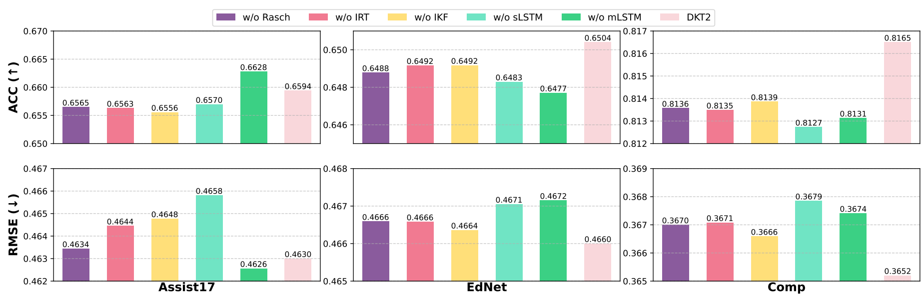

Fig. 9. Ablation study on ACC and RMSE.

Prediction Results with Varying History Lengths Fig. 7 and Fig. 8 show the prediction performance of DKT2 and several representative baselines with different history lengths on the EdNet and Comp, respectively.

Prediction Results Under Three Input Settings Table 8 presents the prediction performance of KT models with weak applicability and comprehensiveness in the last 5 steps on EdNet and Comp under three different input settings.

Ablation Study on ACC and RMSE Fig. 9 shows the ablation study of DKT2 on ACC and RMSE.

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.: Deep graph memory networks for forgetting-robust knowledge tracing. IEEE Transactions on Knowledge and Data Engineering **35**(8), 7844–7855 (2022)
3. Alharthi, M., Mahmood, A.: xlstmtime: Long-term time series forecasting with xlstm. arXiv preprint arXiv:2407.10240 (2024)
4. Alkin, B., Beck, M., Pöppel, K., Hochreiter, S., Brandstetter, J.: Vision-lstm: xlstm as generic vision backbone. arXiv preprint arXiv:2406.04303 (2024)
5. Anderson, J.A., Silverstein, J.W., Ritz, S.A., Jones, R.S.: Distinctive features, categorical perception, and probability learning: Some applications of a neural model. Psychological review **84**(5), 413 (1977)
6. Ba, J., Hinton, G.E., Mnih, V., Leibo, J.Z., Ionescu, C.: Using fast weights to attend to the recent past. Advances in neural information processing systems **29** (2016)
7. Beck, M., Pöppel, K., Spanring, M., Auer, A., Prudnikova, O., Kopp, M., Klambauer, G., Brandstetter, J., Hochreiter, S.: xlstm: Extended long short-term memory. arXiv preprint arXiv:2405.04517 (2024)
8. Cao, Y., Zhang, W.: Mamba4kt: An efficient and effective mamba-based knowledge tracing model. arXiv preprint arXiv:2405.16542 (2024)
9. Chen, J., Liu, Z., Huang, S., Liu, Q., Luo, W.: Improving interpretability of deep sequential knowledge tracing models with question-centric cognitive representations. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 14196–14204 (2023)
10. Chen, M., Guan, Q., He, Y., He, Z., Fang, L., Luo, W.: Knowledge tracing model with learning and forgetting behavior. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 3863–3867 (2022)
11. Cheng, K., Peng, L., Wang, P., Ye, J., Sun, L., Du, B.: Dygkt: Dynamic graph learning for knowledge tracing. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 409–420 (2024)
12. Cheng, S., Liu, Q., Chen, E., Zhang, K., Huang, Z., Yin, Y., Huang, X., Su, Y.: Adaptkt: A domain adaptable method for knowledge tracing. In: Proceedings of the fifteenth ACM international conference on web search and data mining. pp. 123–131 (2022)
13. Cui, C., Ma, H., Zhang, C., Zhang, C., Yao, Y., Chen, M., Ma, Y.: Do we fully understand students' knowledge states? identifying and mitigating answer bias in knowledge tracing. arXiv preprint arXiv:2308.07779 (2023)
14. Cui, C., Yao, Y., Zhang, C., Ma, H., Ma, Y., Ren, Z., Zhang, C., Ko, J.: Dgekt: a dual graph ensemble learning method for knowledge tracing. ACM Transactions on Information Systems **42**(3), 1–24 (2024)
15. Cui, J., Chen, Z., Zhou, A., Wang, J., Zhang, W.: Fine-grained interaction modeling with multi-relational transformer for knowledge tracing. ACM Transactions on Information Systems **41**(4), 1–26 (2023)
16. Cui, J., Qian, H., Jiang, B., Zhang, W.: Leveraging pedagogical theories to understand student learning process with graph-based reasonable knowledge tracing. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 502–513 (2024)

17. Dutta, P., Bose, S., Roy, S.K., Mitra, S.: Are vision xlstm embedded unet more reliable in medical 3d image segmentation? arXiv preprint arXiv:2406.16993 (2024)
18. Elman, J.L.: Finding structure in time. Cognitive science **14**(2), 179–211 (1990)
19. Fu, L., Guan, H., Du, K., Lin, J., Xia, W., Zhang, W., Tang, R., Wang, Y., Yu, Y.: Sinkt: A structure-aware inductive knowledge tracing model with large language model. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. pp. 632–642 (2024)
20. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: Continual prediction with lstm. Neural computation **12**(10), 2451–2471 (2000)
21. 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)
22. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: Lstm: A search space odyssey. IEEE transactions on neural networks and learning systems **28**(10), 2222–2232 (2016)
23. Guan, Q., Duan, X., Bian, K., Chen, G., Huang, J., Gong, Z., Fang, L.: Kvfkt: A new horizon in knowledge tracing with attention-based embedding and forgetting curve integration. In: Proceedings of the 31st International Conference on Computational Linguistics. pp. 4399–4409 (2025)
24. 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)
25. Guo, Y., Shen, S., Liu, Q., Huang, Z., Zhu, L., Su, Y., Chen, E.: Mitigating cold-start problems in knowledge tracing with large language models: An attribute-aware approach. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. pp. 727–736 (2024)
26. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
27. He, L., Li, X., Wang, P., Tang, J., Wang, T.: Man: Memory-augmented attentive networks for deep learning-based knowledge tracing. ACM Transactions on Information Systems **42**(1), 1–22 (2023)
28. Hochreiter, S.: Untersuchungen zu dynamischen neuronalen netzen. Diploma, Technische Universität München **91**(1), 31 (1991)
29. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies (2001)
30. Hochreiter, S., Schmidhuber, J.: Lstm can solve hard long time lag problems. Advances in neural information processing systems **9** (1996)
31. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
32. Huang, C.Q., Huang, Q.H., Huang, X., Wang, H., Li, M., Lin, K.J., Chang, Y.: Xkt: towards explainable knowledge tracing model with cognitive learning theories for questions of multiple knowledge concepts. IEEE Transactions on Knowledge and Data Engineering (2024)
33. 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)
34. Huang, T., Ou, X., Yang, H., Hu, S., Geng, J., Hu, J., Xu, Z.: Remembering is not applying: Interpretable knowledge tracing for problem-solving processes. In:

Proceedings of the 32nd ACM International Conference on Multimedia. pp. 3151–3159 (2024)
35. 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)
36. Kohonen, T.: Correlation matrix memories. IEEE transactions on computers **100**(4), 353–359 (1972)
37. Lee, W., Chun, J., Lee, Y., Park, K., Park, S.: Contrastive learning for knowledge tracing. In: Proceedings of the ACM Web Conference 2022. pp. 2330–2338 (2022)
38. 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
39. Liu, F., Bu, C., Zhang, H., Wu, L., Yu, K., Hu, X.: Fdkt: Towards an interpretable deep knowledge tracing via fuzzy reasoning. ACM Transactions on Information Systems **42**(5), 1–26 (2024)
40. Liu, G., Zhan, H., Kim, J.j.: Question difficulty consistent knowledge tracing. In: Proceedings of the ACM Web Conference 2024. pp. 4239–4248 (2024)
41. Liu, Q., Huang, Z., Yin, Y., Chen, E., Xiong, H., Su, Y., Hu, G.: Ekt: Exercise-aware knowledge tracing for student performance prediction. IEEE Transactions on Knowledge and Data Engineering **33**(1), 100–115 (2019)
42. Liu, Y., Yang, Y., Chen, X., Shen, J., Zhang, H., Yu, Y.: Improving knowledge tracing via pre-training question embeddings. arXiv preprint arXiv:2012.05031 (2020)
43. 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)
44. 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)
45. Long, T., Liu, Y., Shen, J., Zhang, W., Yu, Y.: Tracing knowledge state with individual cognition and acquisition estimation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 173–182 (2021)
46. Long, T., Qin, J., Shen, J., Zhang, W., Xia, W., Tang, R., He, X., Yu, Y.: Improving knowledge tracing with collaborative information. In: Proceedings of the fifteenth ACM international conference on web search and data mining. pp. 599–607 (2022)
47. 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 Web Conference 2024. pp. 4479–4488 (2024)
48. Milakov, M., Gimelshein, N.: Online normalizer calculation for softmax. arXiv preprint arXiv:1805.02867 (2018)
49. Minn, S., Vie, J.J., Takeuchi, K., Kashima, H., Zhu, F.: Interpretable knowledge tracing: Simple and efficient student modeling with causal relations. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 12810–12818 (2022)
50. Minn, S., Yu, Y., Desmarais, M.C., Zhu, F., Vie, J.J.: Deep knowledge tracing and dynamic student classification for knowledge tracing. In: 2018 IEEE International conference on data mining (ICDM). pp. 1182–1187. IEEE (2018)
51. 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)

52. 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)
53. Pandey, S., Karypis, G.: A self-attentive model for knowledge tracing. arXiv preprint arXiv:1907.06837 (2019)
54. Pandey, S., Srivastava, J.: Rkt: relation-aware self-attention for knowledge tracing. In: Proceedings of the 29th ACM international conference on information & knowledge management. pp. 1205–1214 (2020)
55. 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)
56. Pu, Y., Liu, F., Shi, R., Yuan, H., Chen, R., Peng, T., Wu, W.: Elakt: Enhancing locality for attentive knowledge tracing. ACM Transactions on Information Systems **42**(4), 1–27 (2024)
57. Schmidhuber, J.: Learning to control fast-weight memories: An alternative to recurrent nets. accepted for publication in. Neural Computation (1992)
58. Shen, S., Chen, E., Liu, Q., Huang, Z., Huang, W., Yin, Y., Su, Y., Wang, S.: Monitoring student progress for learning process-consistent knowledge tracing. IEEE Transactions on Knowledge and Data Engineering **35**(8), 8213–8227 (2022)
59. Shen, S., Huang, Z., Liu, Q., Su, Y., Wang, S., Chen, E.: Assessing student's dynamic knowledge state by exploring the question difficulty effect. In: Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval. pp. 427–437 (2022)
60. 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)
61. Shen, S., Liu, Q., Chen, E., Wu, H., Huang, Z., Zhao, W., Su, Y., Ma, H., Wang, S.: 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. pp. 1857–1860 (2020)
62. Shen, X., Yu, F., Liu, Y., Liang, R., Wan, Q., Yang, K., Sun, J.: Revisiting knowledge tracing: A simple and powerful model. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 263–272 (2024)
63. Sun, J., Yu, F., Liu, S., Luo, Y., Liang, R., Shen, X.: Adversarial bootstrapped question representation learning for knowledge tracing. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 8016–8025 (2023)
64. Sun, J., Yu, F., Wan, Q., Li, Q., Liu, S., Shen, X.: Interpretable knowledge tracing with multiscale state representation. In: Proceedings of the ACM on Web Conference 2024. pp. 3265–3276 (2024)
65. Tong, H., Wang, Z., Zhou, Y., Tong, S., Han, W., Liu, Q.: Introducing problem schema with hierarchical exercise graph for knowledge tracing. In: Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval. pp. 405–415 (2022)
66. Tong, S., Liu, Q., Huang, W., Hunag, Z., Chen, E., Liu, C., Ma, H., Wang, S.: Structure-based knowledge tracing: An influence propagation view. In: 2020 IEEE international conference on data mining (ICDM). pp. 541–550. IEEE (2020)
67. Vie, J.J., Kashima, H.: Knowledge tracing machines: Factorization machines for knowledge tracing. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 750–757 (2019)

68. 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)
69. Wang, C., Sahebi, S.: Continuous personalized knowledge tracing: Modeling long-term learning in online environments. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. pp. 2616–2625 (2023)
70. Wang, X., Chen, L., Zhang, M.: Deep attentive model for knowledge tracing. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 10192–10199 (2023)
71. Xu, B., Huang, Z., Liu, J., Shen, S., Liu, Q., Chen, E., Wu, J., Wang, S.: Learning behavior-oriented knowledge tracing. In: Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining. pp. 2789–2800 (2023)
72. Yang, S., Liu, X., Su, H., Zhu, M., Lu, X.: Deep knowledge tracing with learning curves. In: 2022 IEEE International Conference on Data Mining Workshops (ICDMW). pp. 282–291. IEEE (2022)
73. Yang, Y., Shen, J., Qu, Y., Liu, Y., Wang, K., Zhu, Y., Zhang, W., Yu, Y.: Gikt: a graph-based interaction model for knowledge tracing. In: Machine learning and knowledge discovery in databases: European conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, proceedings, part I. pp. 299–315. Springer (2021)
74. Yeung, C.K.: Deep-irt: Make deep learning based knowledge tracing explainable using item response theory. arXiv preprint arXiv:1904.11738 (2019)
75. 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)
76. 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)
77. 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)
78. Zhang, M., Zhu, X., Zhang, C., Ji, Y., Pan, F., Yin, C.: Multi-factors aware dual-attentional knowledge tracing. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 2588–2597 (2021)
79. Zhang, M., Zhu, X., Zhang, C., Pan, F., Qian, W., Zhao, H.: No length left behind: Enhancing knowledge tracing for modeling sequences of excessive or insufficient lengths. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. pp. 3226–3235 (2023)
80. Zhang, M., Zhu, X., Zhang, C., Qian, W., Pan, F., Zhao, H.: 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. pp. 3236–3246 (2023)
81. Zhao, S., Sahebi, S.: Graph-enhanced multi-activity knowledge tracing. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 529–546. Springer (2023)
82. Zheng, N., Shan, Z.: Co-attention and contrastive learning driven knowledge tracing. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 177–194. Springer (2024)

83. Zhou, H., Bamler, R., Wu, C.M., Tejero-Cantero, Á.: Predictive, scalable and interpretable knowledge tracing on structured domains. arXiv preprint arXiv:2403.13179 (2024)
84. Zhou, Y., Lv, Z., Zhang, S., Chen, J.: Disentangled knowledge tracing for alleviating cognitive bias. In: Proceedings of the ACM on Web Conference 2025. pp. 2633–2645 (2025)
85. Zhu, Q., Cai, Y., Fan, L.: Seg-lstm: Performance of xlstm for semantic segmentation of remotely sensed images. arXiv preprint arXiv:2406.14086 (2024)