RETRACEKT: RELEVANCE-GUIDED MULTIMODAL STATE-SPACE KNOWLEDGE TRACING

Anonymous authors

ABSTRACT

Knowledge tracing (KT) estimates a learner's evolving knowledge state from a sequence of exercise responses. Two properties of modern learning platforms make this problem harder: essential evidence is often embedded in diagrams rather than text, and long histories contain many interactions that are irrelevant to the next exercise. We present ReTraceKT, a multimodal KT framework that retrieves semantically relevant evidence before tracing knowledge dynamics. ReTraceKT first uses a vision–language model offline to render each exercise image into structured descriptions and aligns them with trainable exercise embeddings. It then builds a thresholded semantic graph over exercises and uses the graph to retrieve a sparse, question-conditioned summary of a learner's history. Finally, a selective state-space tracer combines local response patterns with long-range state transitions in linear sequence complexity. Experiments on NIPS34, MOOC-Radar, and XES3G5M show consistent improvements over strong recurrent, attention-based, and sparse KT baselines. Relative to the strongest baseline, ReTraceKT improves AUC by 3.22%, 1.82%, and 2.08% on the three datasets, respectively; ablations confirm that semantic rendering and relevance retrieval are complementary. Overall, ReTraceKT delivers consistently strong and stable performance across the three datasets.

Keywords: knowledge tracing, multimodal representation, state-space models, relevance retrieval

1 INTRODUCTION

Knowledge tracing (KT) predicts whether a learner will answer a future exercise correctly from their past interactions. The prediction supports exercise recommendation, adaptive assessment, and personalized learning paths. Since Bayesian knowledge tracing (BKT) Corbett & Anderson (1994); Kasurinen & Nikula (2009), the field has progressed from recurrent models such as DKT Piech et al. (2015) and memory networks Zhang et al. (2017) to attention-based models including SAKT Pandey & Karypis (2019) and AKT Ghosh et al. (2020). These approaches are increasingly accurate, but they largely treat an exercise as an identifier, a knowledge-concept tag, or a short text string.

This abstraction misses an important change in online education: exercises now combine prose, formulas, charts, and geometric diagrams. In many mathematics and science questions, the diagram contains quantities and spatial relations absent from the text. A generic image encoder can recognize appearance, but does not necessarily expose the relations needed to distinguish knowledge concepts. At the same time, a learner's history may contain hundreds of interactions. Only a small subset is semantically relevant to the current exercise; dense attention both spends quadratic computation and mixes unrelated evidence into the state estimate.

We argue that these difficulties should be handled at different stages. Content understanding can be performed once per exercise, offline, while personalized tracing must remain lightweight online. This separation motivates ReTraceKT: *render* image semantics, *retrieve* relevant historical interactions, and *trace* the remaining temporal signal. A vision–language model (VLM) produces structured descriptions of the question, options, and visual relations. Their fixed text embeddings augment learned exercise and concept representations. A thresholded exercise graph then selects semantically related history and forms a normalized retrieval summary. A selective state-space model (SSM) propagates the learner state with linear complexity in sequence length.

Our contributions are threefold:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

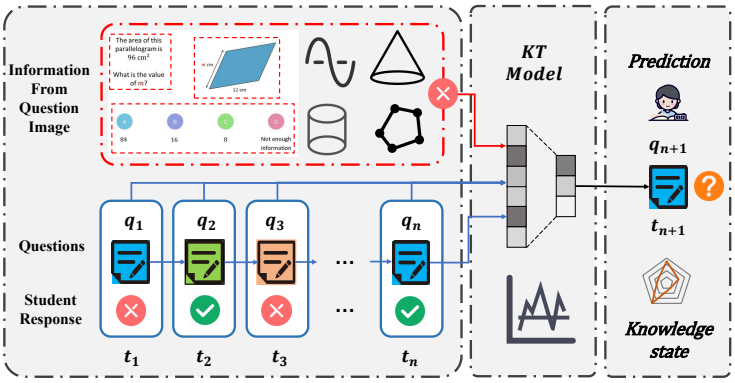

Figure 1: Multimodal knowledge tracing predicts a learner's next response and knowledge state from an interaction history whose exercises may contain text, symbols, and diagrams.

- We formulate multimodal KT as an offline semantic rendering problem followed by online relevance-guided tracing, making rich exercise understanding compatible with low-cost inference.
- We introduce a sparse semantic retrieval operator and a selective state tracer that jointly suppress irrelevant interactions while preserving local and long-range learning dynamics.
- ReTraceKT improves the strongest AUC baseline by 3.22% on NIPS34, 1.82% on MOOC-Radar, and 2.08% on XES3G5M. Ablations identify where these gains originate.

2 Related Work

2.1 Knowledge Tracing

Deep KT replaces hand-crafted latent-state transitions with neural sequence modeling. DKT encodes the complete response history with an LSTM, whereas DKVMN separates static concept keys from value memories updated by learner responses Piech et al. (2015); Zhang et al. (2017). These models provide strong sequential baselines, but they mainly represent exercises through IDs or concept tags and therefore do not directly expose multimodal content.

Attention-based models provide direct access to distant interactions. SAKT queries past responses with the target exercise Pandey & Karypis (2019); AKT adds contextual embeddings and distance decay to model forgetting Ghosh et al. (2020); and RKT injects exercise relations into attention Pandey & Srivastava (2020). SAINT separates exercise and response streams Choi et al. (2020), DTransformer combines question-aware tracing with contrastive regularization Yin et al. (2023), and SimpleKT shows that regularized dot-product attention remains competitive Liu et al. (2023a). Although these methods improve long-range access, dense pairwise comparison can be costly and may assign weight to temporally available but semantically irrelevant events.

Efficiency and robustness methods include sparse attention Huang et al. (2023), anomaly suppression Ma et al. (2024), multi-level knowledge representations Guo et al. (2025), and selective state-space dynamics Cao & Zhang (2024). These methods reduce computation or sensitivity to noisy records, but do not jointly ground diagram semantics and use them for personalized history retrieval. ReTraceKT addresses this gap by grounding multimodal exercises before sparse retrieval and state tracing.

2.2 Semantic Grounding and Efficient Tracing

Vision–language models align visual and textual evidence and can verbalize fine-grained relations that are difficult to recover from a global image vector Achiam et al. (2023); Bai et al. (2025a). Recent educational work uses large models for cognitive representations and teacher-like reasoning Li et al. (2025); Xu et al. (2025). ReTraceKT instead uses a VLM offline to produce cached, inspectable fields, avoiding foundation-model inference for every learner interaction.

Selective state-space models provide input-dependent transitions in linear time, but recurrent compression alone does not explicitly recover a distant, relevant exercise. ReTraceKT combines a sparse

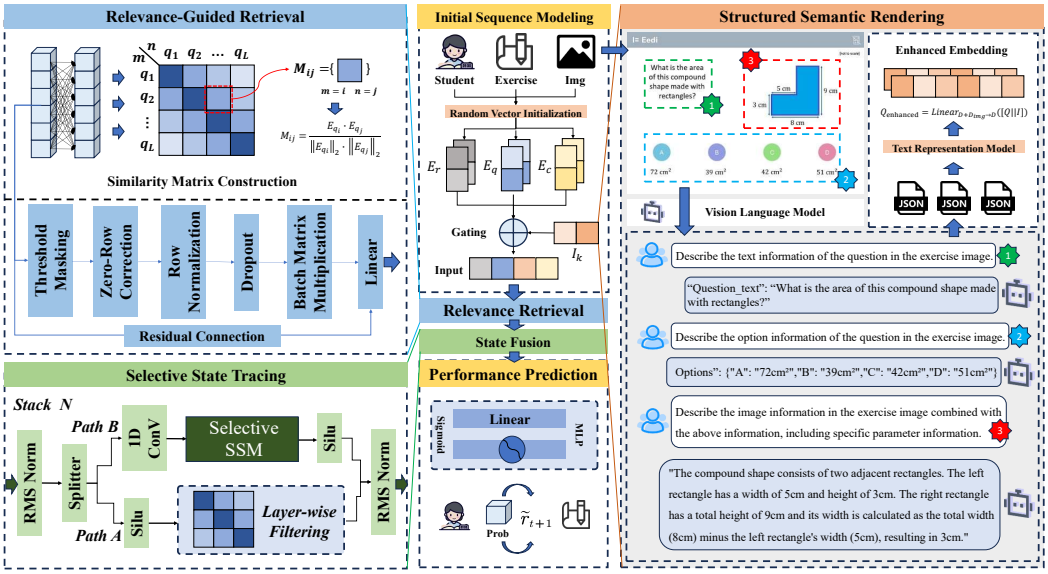

Figure 2: ReTraceKT separates corpus-level multimodal understanding from learner-level inference. Orange components are precomputed; blue/green components form the online tracing model.

semantic graph with selective state propagation: content grounding and evidence selection occur before efficient chronological tracing, while the online prediction path remains lightweight.

3 METHOD

3.1 TASK AND ARCHITECTURE

Let q_t, c_t, and $r_t \in \{0, 1\}$ denote the exercise, its knowledge concept, and the observed response at step t. Given $\mathcal{H}_t = \{(q_i, c_i, r_i)\}_{i=1}^{t}$ and next exercise q_{t+1}, KT estimates $p(r_{t+1} = 1 \mid \mathcal{H}_t, q_{t+1})$. Figure 2 summarizes ReTraceKT. Its semantic renderer is executed once for each exercise; retrieval and state tracing operate on each learner sequence.

This formulation separates three sources of evidence. Exercise content describes what knowledge is tested; the response indicates whether that knowledge was demonstrated; and chronology describes how the learner state evolves. Existing ID-based KT models emphasize the latter two, while direct multimodal fusion adds content without controlling which historical content should influence a prediction. ReTraceKT explicitly connects these sources: rendered content defines the retrieval graph, observed responses populate the retrieved messages, and the state-space tracer preserves their temporal order.

3.2 STRUCTURED SEMANTIC RENDERING

For exercise q, we prompt a VLM in three stages: transcribe question text, transcribe answer options, and describe visual objects, quantities, and relations using the preceding transcription as context. The output is stored as a structured record rather than an unrestricted rationale. A frozen text representation model maps the record to $v_q \in \mathbb{R}^{d_v}$:

$$v_q = \mathrm{TRM}(\mathrm{VLM}(\mathrm{image}_q, \mathrm{prompt})). \tag{1}$$

This pipeline turns implicit geometry or chart structure into a representation comparable across questions. Because v_q is cached, VLM latency is absent at training and inference time.

Let $e_q, e_c, e_r \in \mathbb{R}^d$ be trainable embeddings. We align the fixed semantic vector with the KT space and construct an interaction representation

$$z_t = W_a[e_{q_t} \| v_{q_t}] + e_{c_t} + e_{r_t}, \tag{2}$$

where $W_a \in \mathbb{R}^{d \times (d+d_v)}$ and $\|$ is concatenation. The next exercise is encoded identically but without a response embedding.

More explicitly, the renderer produces three fields: question transcription s_q^{ques}, answer options s_q^{opt}, and grounded relations s_q^{rel}. We encode them independently and retain their complementary evidence through a learned fusion gate:

$$v_q^{(k)} = \text{TRM}(s_q^{(k)}), \quad k \in \{\text{ques}, \text{opt}, \text{rel}\}. \tag{3}$$

$$\gamma_q = \text{softmax}\Big(W_\gamma[v_q^{(\text{ques})} \| v_q^{(\text{opt})} \| v_q^{(\text{rel})}]\Big). \tag{4}$$

$$v_q = \text{LN}\left(\sum_k \gamma_q^{(k)} W_k v_q^{(k)}\right). \tag{5}$$

The gate prevents a long transcription from overwhelming short but decisive geometric relations. All v_q are computed once and cached; only W_a, the field projections, and the downstream tracer are optimized with KT supervision.

3.3 Relevance-Guided History Retrieval

Dense attention implicitly learns relevance from interaction labels. We instead expose exercise relevance using the rendered semantics. For exercises q_i, q_j, define a corpus-level graph

$$S_{ij} = \frac{v_{q_i}^\top v_{q_j}}{\|v_{q_i}\|_2 \|v_{q_j}\|_2}, \qquad A_{ij} = S_{ij}\, \mathbb{I}[S_{ij} \geq \tau]. \tag{6}$$

For every row with no retained neighbor, we set its diagonal entry to one. Row normalization gives $\bar{A}_{ij} = A_{ij}/(\sum_k A_{ik} + \epsilon)$. Given a learner subsequence, the corresponding graph slice retrieves a sparse semantic summary:

$$\tilde{z}_t = z_t + W_g \sum_{j \leq t} \bar{A}_{q_t q_j} z_j. \tag{7}$$

The causal restriction $j \leq t$ prevents future leakage. Unlike top-k attention, the threshold allows the number of retrieved interactions to adapt to question density. It also yields a direct explanation: retained historical questions are those connected to the query in semantic space.

The graph is constructed only from exercise content and never from response labels. It can therefore be reused across learners without leaking performance information. For a minibatch, we gather the rows and columns indexed by its exercise IDs rather than materializing a learner-specific dense graph. If E_τ denotes the retained edge set, storage is $O(|E_\tau|)$ in sparse form. The residual term in Eq. equation 7 protects unique exercises whose neighborhood contains little useful evidence, while row normalization prevents high-degree questions from dominating merely because they have more neighbors.

For clarity, let $Z = [z_1, \ldots, z_L]^\top$ and let $P_t = \{j \mid j \leq t,\, A_{q_t q_j} > 0\}$ be the causal semantic neighborhood of step t. The learner-specific sparse operator is

$$G_{tj} = \frac{A_{q_t q_j} \mathbb{I}[j \in P_t]}{\sum_{k \in P_t} A_{q_t q_k} + \epsilon}, \qquad \widetilde{Z} = Z + G W_g Z. \tag{8}$$

Thus each row of G sums to at most one, contains no future interaction, and can be implemented by indexed sparse aggregation. Its cost is $O(|E_\tau^{(u)}|d)$ for learner u, rather than $O(L^2 d)$ dense attention.

3.4 Selective State Tracing

The retrieved sequence enters N residual state-space blocks. After RMS normalization, an input projection is split into a content stream x and gate g. A depthwise convolution extracts short response motifs. The content stream then parameterizes a selective SSM:

$$x'_t = \text{SiLU}(\text{DWConv}(x_{1:t})). \tag{9}$$

$$(\Delta_t, B_t, C_t) = W_s x'_t. \tag{10}$$

$$\bar{A}_t = \exp(\Delta_t A), \qquad \bar{B}_t = \Delta_t B_t. \tag{11}$$

$$h_t = \bar{A}_t h_{t-1} + \bar{B}_t x'_t. \tag{12}$$

$$y_t = C_t h_t + D x'_t. \tag{13}$$

Table 1: Statistics of the processed datasets.

	NIPS34	MOOC-Radar	XES3G5M
Students	9,401	14,224	18,066
Exercises	948	2,510	7,652
Concepts	57	1,467	865
Interactions	1,399,470	897,229	5,549,635
Avg. length	149	63	307

A is initialized with stable negative eigenvalues; Δ_t is made positive using softplus. The block output is $W_o(y_t \odot \mathrm{SiLU}(g_t))$ plus a residual connection. We reapply Eq. equation 7 between blocks, allowing semantic neighbors to exchange progressively refined states. The SSM costs $O(Ld)$ for sequence length L, while the graph is precomputed and its online cost depends on retained edges.

Writing $H^{(0)} = Z$, the complete layer-wise update is

$$R^{(\ell)} = H^{(\ell-1)} + G^{(u)} W_g^{(\ell)} H^{(\ell-1)}, \tag{14}$$

$$U^{(\ell)} = \mathrm{SSM}_\ell(\mathrm{RMSNorm}(R^{(\ell)})), \tag{15}$$

$$H^{(\ell)} = R^{(\ell)} + W_o^{(\ell)} \left(U^{(\ell)} \odot \mathrm{SiLU}(g^{(\ell)}) \right). \tag{16}$$

This formulation makes the model logic explicit: semantic retrieval first selects relevant evidence, the selective recurrence propagates it in chronological order, and the residual path preserves the original interaction signal.

The two branches play different roles. Depthwise convolution emphasizes short motifs such as consecutive successes or repeated failure, while the recurrent state retains evidence over longer gaps. The input-dependent Δ_t, B_t, and C_t allow the transition to change with the exercise and response instead of applying one fixed forgetting rate. Layer-wise retrieval then gives the state tracer another opportunity to incorporate a distant but semantically related interaction after local temporal features have been refined.

For next-response prediction, the final learner state u_t is combined with the next-exercise representation and passed through a two-layer MLP:

$$\hat{r}_{t+1} = \sigma\big(w^\top \mathrm{GELU}(W_p[u_t \| z_{t+1}] + b_p) + b\big). \tag{17}$$

All online parameters are trained end-to-end with masked binary cross entropy and weight decay:

$$\mathcal{L} = -\frac{1}{|\Omega|} \sum_{t \in \Omega} [r_{t+1} \log \hat{r}_{t+1} + (1 - r_{t+1}) \log(1 - \hat{r}_{t+1})] + \lambda \|\Theta\|_2^2, \tag{18}$$

where Ω excludes padding positions and Θ contains all trainable online parameters. The cached VLM outputs are fixed, so gradients never pass through the rendering model.

4 Experiments

We organize the evaluation around four questions: **RQ1**, does ReTraceKT improve next-response prediction across datasets of different scales? **RQ2**, how much do semantic rendering and relevance retrieval contribute individually and jointly? **RQ3**, is performance robust to sequence and architectural hyperparameters? **RQ4**, does relevance-guided state tracing provide interpretable and efficient long-sequence inference?

4.1 Setup

Datasets. We evaluate on three public multimodal datasets. NIPS34 contains school mathematics exercises with images Wang et al. (2020); MOOC-Radar covers fine-grained interactions from an online course platform Yu et al. (2023); XES3G5M is a large-scale tutoring dataset Liu et al. (2023b). Table 1 reports the processed splits used in all experiments.

Protocol. We compare recurrent, attention-based, sparse/robust, and recent competitive KT models listed in Table 2, using official settings when available. Learners are split 80/20 without cross-split histories; interactions are chronological, invalid records are removed, and long histories are partitioned

Table 2: Predictive performance and probability error. Best results are bold and strongest baselines are underlined.

Model	NIPS34				MOOC-Radar				XES3G5M			
	AUC	ACC	RMSE	MAE	AUC	ACC	RMSE	MAE	AUC	ACC	RMSE	MAE
DKT Piech et al. (2015)	0.7624	0.6843	0.4772	0.2793	0.8342	0.8620	0.3723	0.1582	0.7843	0.8226	0.3964	0.3623
DKVMN Zhang et al. (2017)	0.7673	0.6935	0.4612	0.2813	0.8440	0.8473	0.3672	0.1412	0.7764	0.8147	0.3932	0.3592
SAKT Pandey & Karypis (2019)	0.7628	0.6774	0.4692	0.2832	0.8365	0.8536	0.3712	0.1632	0.7582	0.8093	0.4022	0.3622
AKT Ghosh et al. (2020)	0.7680	0.6947	0.4620	0.2812	<u>0.8503</u>	0.8625	<u>0.3513</u>	0.1452	0.7902	0.8164	<u>0.3912</u>	0.3602
RKT Pandey & Srivastava (2020)	0.7626	0.7023	0.4562	0.2892	0.8434	0.8592	0.3682	<u>0.1372</u>	0.7795	0.8056	0.3981	0.3587
SAINT Choi et al. (2020)	0.7594	0.6740	0.4712	0.2932	0.8423	0.8394	0.3630	0.1592	0.7627	0.8113	0.3971	0.3612
SparseKT Huang et al. (2023)	0.7115	0.6920	0.4722	0.2933	0.8024	0.8396	0.3772	0.1512	0.7263	0.8065	0.3962	0.3613
ExtraKT Li et al. (2024)	0.7742	0.7073	0.4600	0.2770	0.8424	0.8623	0.3642	0.1382	0.7825	0.8152	0.3922	<u>0.3582</u>
MIKT Sun et al. (2024)	0.7813	0.7146	0.4517	0.2871	0.8467	0.8579	0.3618	0.1420	0.7874	0.8109	0.3944	0.3634
HD-KT Ma et al. (2024)	0.7786	0.7049	0.4678	0.3017	0.8340	0.8476	0.3768	0.1439	0.7812	0.8094	0.3986	0.3681
LefoKT Bai et al. (2025b)	0.7683	0.6952	0.4613	0.3013	0.8353	<u>0.8652</u>	0.3890	0.1483	0.7910	<u>0.8234</u>	<u>0.3912</u>	0.3592
RobustKT Guo et al. (2025)	0.7739	0.7192	0.4691	0.2862	0.8312	0.8519	0.3736	0.1417	0.7849	0.8119	0.3928	0.3674
KeenKT Li et al. (2026)	<u>0.7891</u>	<u>0.7298</u>	<u>0.4487</u>	<u>0.2762</u>	0.8361	0.8573	0.3641	0.1409	<u>0.7917</u>	0.8204	0.3917	0.3594
ReTraceKT	**0.8145**	**0.7453**	**0.4264**	**0.2563**	**0.8658**	**0.8749**	**0.3462**	**0.1252**	**0.8082**	**0.8269**	**0.3875**	**0.3555**

into consecutive 200-step windows. The VLM sees only public exercise content, so cached semantics and graph edges are independent of learner responses and splits.

Training and metrics. We use Adam with $d = 128$, $\tau = 0.2$, state size 8, validation-AUC early stopping, and dropout on retrieval weights and residual blocks. All methods use the same processed splits, and results report mean $\pm$ standard deviation across repeated runs. AUC/ACC measure ranking and thresholded correctness; RMSE/MAE measure probability error. Higher AUC/ACC and lower RMSE/MAE are better.

4.2 Main Results

RQ1: Overall effectiveness. We compare ranking, classification, and probability error across datasets of substantially different scales. Table 2 shows that ReTraceKT leads every dataset in AUC and ACC while also achieving the lowest RMSE and MAE, so its ranking gains are not obtained by sacrificing probability quality.

Table 2 shows that ReTraceKT obtains the best AUC and ACC on all datasets. Relative to the strongest baseline per column, its AUC increases by 3.22% on NIPS34, 1.82% on MOOC-Radar, and 2.08% on XES3G5M. Gains are largest on NIPS34, whose diagram-rich mathematics questions particularly benefit from structured rendering. MOOC-Radar already has strong concept signals, but retrieval still improves AUC and ACC. The consistent XES3G5M gain indicates that the approach scales beyond small exercise vocabularies.

ReTraceKT obtains RMSE values of 0.4264, 0.3462, and 0.3875 and MAE values of 0.2563, 0.1252, and 0.3555 on NIPS34, MOOC-Radar, and XES3G5M, respectively. Compared with the best baseline, the largest improvements are NIPS34 RMSE (4.97%) and MOOC-Radar MAE (8.75%). Thus, higher ranking performance does not come at the cost of poorly calibrated probabilities.

Dense attention is not consistently better than recurrent baselines, while sparsification without content-aware semantics can remove useful interactions. ReTraceKT derives its graph from multimodal content and couples it to an efficient temporal backbone; its small standard deviations indicate stable improvements.

4.3 Ablation and Sensitivity

RQ2: Component contribution. We isolate semantic rendering and relevance retrieval under identical training settings. The ablations test whether either component alone explains the gain and whether their combination provides complementary evidence.

We remove structured semantic rendering (SR), relevance-guided retrieval (RR), or both. Without SR, exercise representations fall back to learned ID/concept embeddings; without RR, the state tracer receives the unfiltered sequence. Table 3 shows that both components help on every dataset. SR contributes more, but RR consistently adds 0.0044–0.0062 AUC over an otherwise complete model.

Table 3: Ablation results for structured semantic rendering (SR) and relevance-guided retrieval (RR).

	NIPS34		MOOC		XES	
Variant	AUC	ACC	AUC	ACC	AUC	ACC
Full	**0.8145**	**0.7453**	**0.8658**	**0.8749**	**0.8082**	**0.8269**
w/o SR	0.7945	0.7293	0.8553	0.8676	0.7916	0.8178
w/o RR	0.8083	0.7354	0.8614	0.8712	0.8024	0.8218
w/o both	0.7817	0.7210	0.8519	0.8651	0.7890	0.8154

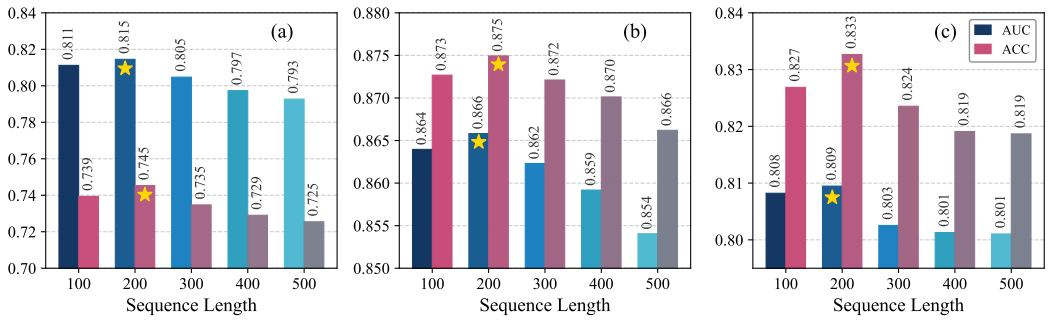

Figure 3: Effect of maximum interaction length on NIPS34, MOOC-Radar, and XES3G5M. Moderate histories provide sufficient learning evidence without excessive unrelated interactions.

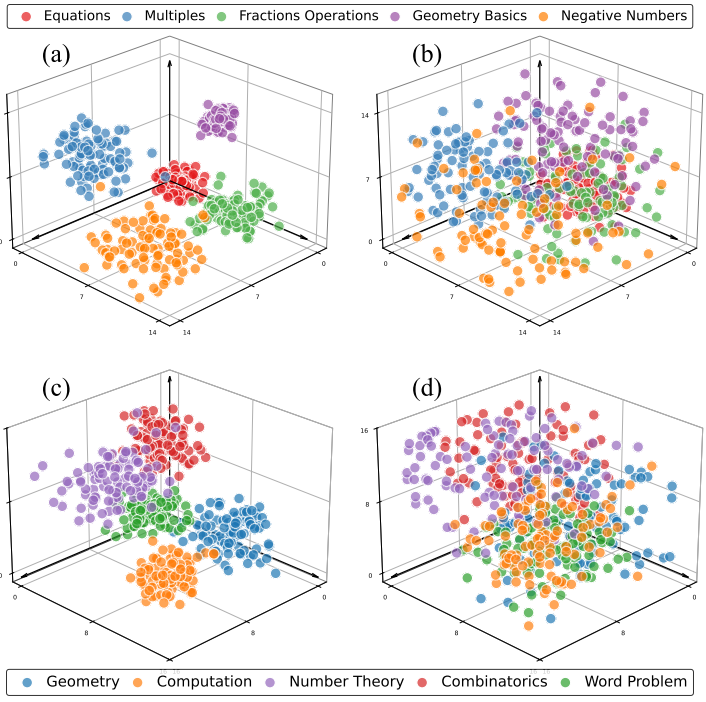

Figure 4: Three-dimensional projections of exercise representations. Structured semantic rendering is shown in the left subplots (a) and (c), which separate knowledge categories more clearly than the direct visual encoding shown in the right subplots (b) and (d).

Removing both produces a larger drop than either isolated removal, supporting the intended coupling: richer semantics improve the graph used for retrieval.

RQ3: Robustness and sensitivity. The retrieval threshold controls the precision–coverage tradeoff. Figure 5 shows a consistent optimum near $\tau = 0.2$ on all three datasets. With no or weak thresholding, the graph admits many marginally related interactions and dilutes the retrieved evidence. Increasing

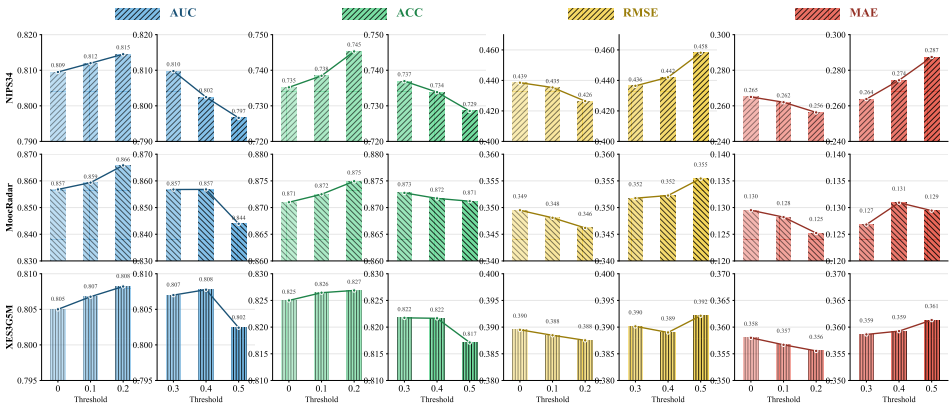

Figure 5: Sensitivity to the semantic-graph threshold on NIPS34, MOOC-Radar, and XES3G5M. A moderate threshold ($\tau = 0.2$) balances useful neighbors and irrelevant edges.

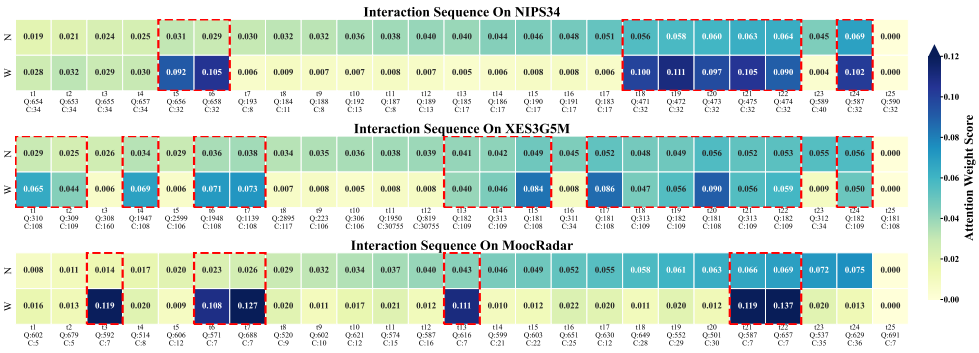

Figure 6: Retrieved weights for representative 24-step histories. Thresholded retrieval concentrates mass on semantically related exercises, while the threshold-free variant distributes weight more diffusely.

τ initially improves both AUC and ACC by removing these noisy edges, whereas overly large values discard useful semantic neighbors and reduce performance. The agreement across datasets indicates that the selected threshold reflects a stable relevance–coverage balance rather than a dataset-specific tuning artifact.

Sequence length. We additionally vary the maximum history from 100 to 500. NIPS34 and XES3G5M gradually lose accuracy as the window becomes very long, consistent with the accumulation of weakly related interactions. MOOC-Radar peaks near 200 steps and then declines. Importantly, ReTraceKT remains competitive throughout the range: retrieval limits the effect of added noise, while the state-space recurrence avoids a quadratic increase in temporal modeling cost. We therefore use 200 as a common operating point rather than independently tuning a favorable length for each dataset.

Representation geometry. Figure 4 shows that rendered vectors form tighter concept clusters than direct visual encodings, especially for geometry and arithmetic concepts in NIPS34. This supports the role of SR in improving exercise representations and retrieval neighbors.

4.4 Efficiency and Interpretation

RQ4: Efficiency and interpretability. We examine both computational scaling and the evidence selected for individual predictions. Runtime measures whether the state-space design remains practical for long histories, while the retrieval heatmaps test whether high-weight interactions are semantically coherent rather than merely recent.

Figure 7 reports online running time at sequence lengths 100, 200, 400, and 800. ReTraceKT requires 1.43, 2.86, 5.63, and 12.62 ms, is second only to DKT, and at length 800 is 2.61× faster than DKVMN

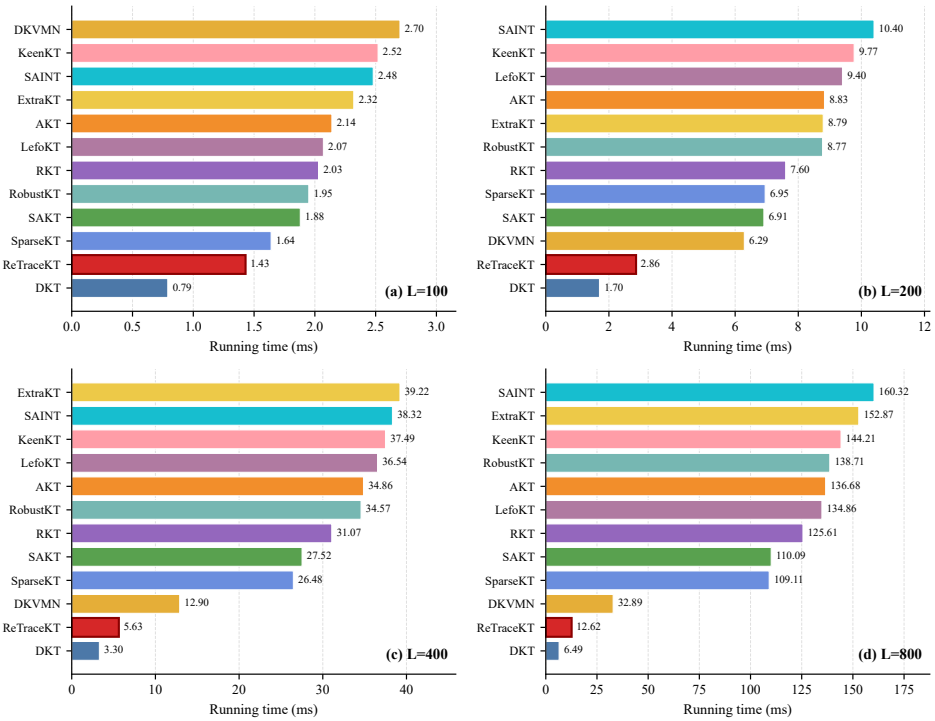

Figure 7: Running-time comparison on NIPS34 at sequence lengths 100, 200, 400, and 800. Only models that both occur in Table 2 and have measurements in the original timing experiment are shown.

and 8.64–12.70× faster than measured attention alternatives. This scaling reflects the linear selective scan.

Semantic rendering is performed once per distinct exercise, then cached. The learner-facing path therefore uses embedding lookup, sparse retrieval, and selective scanning; its cost depends on interactions and retained graph edges rather than VLM size.

Retrieved histories provide a qualitative explanation: NIPS34 targets concentrate weights on earlier concept-32 exercises, while an XES3G5M parity target retrieves semantically adjacent odd/even exercises rather than only exact matches.

Without thresholding, weights become diffuse and overly recent; Figure 6 shows that thresholded retrieval instead selects older, concept-relevant interactions. These weights indicate model focus rather than causality but remain useful for diagnosis.

5 Conclusion

We introduced ReTraceKT, a relevance-guided multimodal knowledge tracing model that separates offline content understanding from online state inference. It converts exercise images into structured semantic fields, aligns them with trainable exercise and concept embeddings, and uses a thresholded semantic graph to retrieve relevant history before selective state-space tracing. Across NIPS34, MOOC-Radar, and XES3G5M, ReTraceKT improves prediction over recurrent, attention-based, and sparse baselines. Ablations confirm the complementarity of semantic rendering and relevance retrieval, while runtime results support the efficiency of the offline-online design.

However, because ReTraceKT relies on multimodal information to represent exercise content and guide historical retrieval, its evaluation is constrained by the limited image information available in many existing KT datasets. Future work should therefore develop and use more diverse public datasets with richer image-text interaction records and broader subject coverage.

AI USE STATEMENT

In this work, we used generative AI tools to assist with generating data-processing code and improving code formatting for readability. All LLM-generated code was reviewed, verified, and tested by the authors for correctness. We reviewed all AI-assisted work and take responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, sensitive personal data, interventions, or other activities that raise ethical concerns. To the best of our knowledge, it does not involve violations of applicable ethical standards.

REPRODUCIBILITY STATEMENT

The algorithm implementation and experimental code are available through the anonymous, downloadable repository `https://anonymous.4open.science/r/ReTraceKT-ABCB`. The repository also includes the dataset processing instructions and related materials needed to reproduce the experiments.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Youheng Bai, Xueyi Li, Zitao Liu, Yaying Huang, Mi Tian, and Weiqi Luo. Rethinking and improving student learning and forgetting processes for attention based knowledge tracing models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 27822–27830, 2025b.

Yang Cao and Wei Zhang. Mamba4kt: An efficient and effective mamba-based knowledge tracing model. *arXiv preprint arXiv:2405.16542*, 2024.

Youngduck Choi, Youngnam Lee, Junghyun Cho, Jineon Baek, Byungsoo Kim, Yeongmin Cha, Dongmin Shin, Chan Bae, and Jaewe Heo. 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.

Albert T. Corbett and John R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1994.

Aritra Ghosh, Neil Heffernan, and Andrew S. Lan. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2330–2339, 2020.

Teng Guo, Yu Qin, Yubin Xia, Mingliang Hou, Zitao Liu, Feng Xia, and Weiqi Luo. Enhancing knowledge tracing through decoupling cognitive pattern from error-prone data. In *Proceedings of the ACM on Web Conference 2025*, pp. 5108–5116, 2025.

Shuyan Huang, Zitao Liu, Xiangyu Zhao, et al. 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.

Jussi Kasurinen and Uolevi Nikula. Estimating programming knowledge with bayesian knowledge tracing. *SIGCSE Bull.*, 41(3):313–317, 2009.

Linqing Li, Zhifeng Wang, et al. Llm supporting knowledge tracing leveraging global subject and student specific knowledge graphs. *Information Fusion*, pp. 103577, 2025.

Xueyi Li, Youheng Bai, Teng Guo, Ying Zheng, Mingliang Hou, Bojun Zhan, Yaying Huang, Zitao Liu, Boyu Gao, and Weiqi Luo. Extending context window of attention based knowledge tracing models via length extrapolation. In *ECAI 2024*, pp. 1479–1486. IOS Press, 2024.

Zhifei Li, Lifan Chen, Jiali Yi, Xiaoju Hou, Yue Zhao, Wenxin Huang, Miao Zhang, Kui Xiao, and Bing Yang. Keenkt: Knowledge mastery-state disambiguation for knowledge tracing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19285–19293, 2026.

Zitao Liu, Qiongqiong Liu, Jiahao Chen, et al. simplekt: a simple but tough-to-beat baseline for knowledge tracing. *arXiv preprint arXiv:2302.06881*, 2023a.

Zitao Liu, Qiongqiong Liu, Teng Guo, Jiahao Chen, Shuyan Huang, Xiangyu Zhao, Jiliang Tang, Weiqi Luo, and Jian Weng. Xes3g5m: A knowledge tracing benchmark dataset with auxiliary information. *Advances in Neural Information Processing Systems*, 36:32958–32970, 2023b.

Haiping Ma, Yong Yang, Chuan Qin, Xiaoshan Yu, Shangshang Yang, Xingyi Zhang, and Hengshu Zhu. Hd-kt: advancing robust knowledge tracing via anomalous learning interaction detection. In *Proceedings of the ACM Web Conference 2024*, pp. 4479–4488, 2024.

S. Pandey and G Karypis. A self-attentive model for knowledge tracing. *In 12th International conference on educational data mining*, pp. pp.384–389, 2019.

Shalini Pandey and Jaideep Srivastava. Rkt: Relation-aware self-attention for knowledge tracing. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1205–1214, 2020.

Chris Piech, Jonathan Bassen, Jonathan Huang, et al. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, volume 28, pp. 505–513, 2015.

Jianwen Sun, Fenghua Yu, Qian Wan, Qing Li, Sannyuya Liu, and Xiaoxuan Shen. Interpretable knowledge tracing with multiscale state representation. In *Proceedings of the ACM Web Conference 2024*, pp. 3265–3276, 2024.

Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yuying Chen, Yu Yin, Zai Huang, and Shijin Wang. Neural cognitive diagnosis for intelligent education systems. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 6153–6161, 2020.

Lixiang Xu, Xianwei Ding, Xin Yuan, Zhanlong Wang, Lu Bai, Enhong Chen, Philip S Yu, and Yuanyan Tang. Improving question embeddings with cognitive representation optimization for knowledge tracing. *IEEE Transactions on Cybernetics*, 2025.

Yu Yin, Le Dai, Zhenya Huang, Shuanghong Shen, Fei Wang, Qi Liu, Enhong Chen, and Xin Li. Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer. In *Proceedings of the ACM Web Conference 2023*, pp. 855–864, 2023.

Jifan Yu, Mengying Lu, Qingyang Zhong, Zijun Yao, Shangqing Tu, Zhengshan Liao, Xiaoya Li, Manli Li, Lei Hou, Hai-Tao Zheng, et al. Moocradar: A fine-grained and multi-aspect knowledge repository for improving cognitive student modeling in moocs. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2924–2934, 2023.

Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 765–774, 2017.