An AI-Powered Evaluation: Understanding which Knowledge Tracing Models Work Best in which Contexts

Anonymous Author(s)
Affiliation
Address
email

Abstract

Knowledge tracing (KT) models a learner's evolving mastery from interaction logs and underpins personalization in tutors, practice systems, and learning analytics. Over three decades, many KT models have been proposed; however, performance varies by dataset characteristics for which the models are trained on, so a model that excels in one setting may under-perform in another. In this work, conducted by an LLM and conceptualized through human-LLM partnership, we explore this phenomenon by conducting a structured synthesis of 124 KT papers spanning classic probabilistic, generalized logistic/factorization, deep sequence, attention/transformer, graph-based, and LLM-augmented approaches (with each paper proposing one or more new models or variants). For each study, we extract key information, including modeling idea, data setting, and outcomes, then code them along eight key contextual dimensions (data scale; sequence length; structure availability: concept-item relations; temporal irregularity/forgetting cues; modality: binary vs. text/code/dialogue; cohort heterogeneity; cold-start/unseen items; interpretability/operational constraints). We apply a two-stage aggregation: (1) within-paper ranking of models on the authors' primary metrics, and (2) context-level win rates/median ranks with quality weights favoring student-wise, chronological, and out-of-distribution protocols, with sensitivity checks for robustness. We find attention/transformers lead on large, long-history logs; graph/dynamic-graph KT dominates when reliable (static or evolving) structure is available; Hawkes/spacing-aware methods win when timing and forgetting matter; LLM/semantic KT excels on text/code/dialogue and improves unseen-item generalization; mixture-of-experts helps in heterogeneous cohorts; and generalized logistic/factorization families remain competitive, interpretable choices in data-constrained settings. We highlight common evaluation pitfalls and synthesize context-dependent patterns across models and datasets, providing practical guidance for context-aware KT model selection.

1 Introduction

Knowledge tracing (KT) models a learner's evolving mastery from their interaction history—e.g., which problems they attempted, whether they were correct, and in what order. By estimating latent knowledge and predicting future performance, KT supports key functions in intelligent tutoring systems and learning analytics: such as adaptive practice [1], timely feedback [2], mastery-based progression [3], and detecting that a student is struggling without making progress [4, 5].

Over the last three decades, KT has expanded from probabilistic and logistic formulations (e.g., BKT, AFM/PFA; [1, 6, 7]) to neural sequence and memory models [8, 9], attention/Transformer variants [10, 11, 12], graph-structured approaches [13], and, more recently, LLM-augmented methods that incorporate item text, code, or dialogue [14, 15]. This progression reflects both methodological advances and the changing priorities of the field: early models emphasized interpretability and mastery estimation, while more recent neural and content-augmented approaches have focused on capturing complex temporal patterns and leveraging multi-modal signals to predict future performance.

KT models are typically trained and evaluated on different datasets that vary in learner demographics, domains, and sampling characteristics (e.g., number of interactions, students, skills, and per-skill practice [16]). Benchmark corpora such as ASSISTments 2009/2012/2017 (K–12 math; [17, 18]), Statics2011 (engineering problem-solving; [9, 19]), and EdNet (a large-scale multi-year platform log with over 100 million interactions; [20]) have become standard testbeds, alongside additional datasets from platforms like Junyi Academy [21], Duolingo [22], and Khan Academy [8]. These corpora differ not only in scale and subject matter but also in sequence length, temporal regularity, structure availability, and cohort composition.

Because of this heterogeneity in datasets, models can perform differentially across datasets, so a method that excels in one setting may under-perform in another [23, 24]. This motivates our main research question: ***Which KT models work best in which contexts?*** In this paper, context is defined as the salient properties of the learning data and deployment setting, including (i) data scale and sequence length; (ii) availability and stability of concept–item structure; (iii) temporal irregularity and forgetting dynamics; (iv) modality (binary correctness versus text/code/dialogue); (v) cohort heterogeneity; (vi) prevalence of cold-start or unseen items; and (vii) requirements around interpretability, robustness, and calibration.

In the past, several studies have tackled this question empirically by comparing multiple KT families across datasets. For example, Gervet et al. [23] ran an extensive comparison on nine real-world corpora and found that logistic regression with appropriate features tends to lead on moderate-sized datasets or when each student has many interactions, whereas DKT (deep learning) leads on very large datasets or when precise temporal information is crucial; classical Markov-process models like BKT generally lag. While valuable, such efforts cover only a slice of today's rapidly expanding model space and dataset conditions, motivating a broader, context-sensitive synthesis.

2 Current Study

Given the expanding space of models and corpora, it is impractical to re-implement and exhaustively benchmark every variant across all datasets. Therefore, we take a systematic approach: we collect and synthesize KT models and variants proposed over the past decades, summarize their data settings and reported outcomes, and analyze how results align with context. Specifically, in the current work, we conduct a structured synthesis of 124 KT papers spanning classic, neural, graph-based, and LLM-augmented approaches (each proposing one or more new KT models or variants). For each study, we extract the modeling idea, data setting, and outcomes, then code them along key contextual dimensions (data scale; sequence length; structure availability/dynamics; temporal irregularity; modality; heterogeneity; cold-start; interpretability/operational constraints). We aggregate evidence using within-paper rankings and context-level win rates with quality-aware weighting and sensitivity checks, yielding insight on which model families tend to work best under which conditions.

3 Methods

3.1 Corpus Construction and Scope

A structured literature synthesis was conducted to identify models that estimate a learner's evolving knowledge state from interaction logs (knowledge tracing, KT). The unit of analysis is a model paper (including major variants) that proposes, extends, or rigorously compares KT approaches in educational data mining, learning analytics, or student modeling venues.

Sources and time window. Digital libraries and preprint servers (major ACM/IEEE venues, Springer/Elsevier journals, `arXiv`) were searched for works published between January 1994 ((which corresponds to the introduction of BKT by [1])and May 2025. References were also snowballed from seed papers (e.g., BKT, AFM/PFA, DKT, DKVMN, SAKT/AKT/SAINT, KTM, SPARFA/Trace). Snowballing, as a method for systematic review, refers to backward- and forward-citation chaining, whereby the reference lists of included papers are screened (backward) and works citing those papers are identified (forward) to surface additional KT models and variants. Iteration continued until further additions yielded diminishing returns. The resulting corpus comprises 124 distinct models/variants.

Inclusion and exclusion criteria. Papers were included if they: (i) proposed a KT model or a substantive KT variant; (ii) evaluated on learner–item interaction data with time order; and (iii) reported at least one predictive metric (e.g., AUC-ROC, log loss, accuracy, F1, κ) against one or more baselines. We retained models focused on specific modalities (e.g., programming/code, dialogue) and on related goals (e.g., dropout prediction) when KT was a central outcome or module. We excluded: (i) purely theoretical notes without empirical evaluation; (ii) items focused solely on static concept discovery without temporal prediction, as the research question targets methods that estimate changes over time in learners' knowledge states, rather than static concept inference; and (iii) duplicate pre-prints of the same model without new experiments (keeping the most complete version).

3.2 Screening, De-duplication, and Data Extraction

Two passes were applied: (1) title/abstract screening; and (2) full-text screening. Records were de-duplicated by title, DOI, or `arXiv` ID, and, when necessary, by author–venue–year. For each included paper, the following information was extracted:

- **Model metadata:** model name/acronym; year; venue; model family (e.g., probabilistic/BKT-like, generalized logistic, factorization, RNN/LSTM, attention/Transformer, memory networks, graph/heterogeneous, contrastive/self-supervised, LLM/semantic, mixture-of-experts, uncertainty-aware).
- **Operational summary:** one sentence describing the core mechanism (e.g., self-attention over past interactions with forgetting bias).
- **Data context:** dataset names; domain (e.g., math, programming); scale (students, items, skills, interactions); sequence length (median/mean if provided); modality (e.g., binary correctness, text, code, dialogue); demographics if reported.
- **Evaluation protocol:** split type (student-wise vs. interaction-wise; chronological vs. random); number of folds/runs; hyperparameter search method.
- **Outcomes:** metrics and values (e.g., AUC-ROC, accuracy, F1, κ, log loss when available); whether improvements were statistically tested; compute cost if reported.

All fields were stored in a spreadsheet [LINK ANONYMIZED] and normalized where feasible (e.g., consistent metric naming, venue/year format).

3.3 Context Taxonomy

To answer which models work best in which contexts, each paper was coded along eight dimensions that plausibly moderate performance. The dimensions, categories, and corresponding examples are presented in Table 1.

When information was missing, values were imputed conservatively from public dataset documentation (e.g., EdNet is large/long; ASSISTments 2009 is small-to-medium with short-to-medium sequences). Conservative imputation prioritized lower-bound categories and stricter uncertainty to reduce the risk of overstating model suitability or inflating win rates; when multiple ranges were plausible, the least favorable category consistent with the documentation was selected to minimize bias in context-level aggregation. Ambiguous cases were coded as unknown and excluded from context-specific tallies.

Table 1: Context taxonomy: dimensions, categories, definitions, and examples.

Dimension / Context	**Category**	**Definition**	**Examples**
Data scale (total interactions)	Small	$< 10^5$ interactions	Single course/semester; pilot study
	Medium	10^5–10^6 interactions	ASSISTments 2009/2012; several classes

Table 1 (continued): Context taxonomy.

Dimension / Context	Category	Definition	Examples
	Large	$> 10^6$ interactions	EdNet-scale platforms; nationwide apps
Sequence length (median per student)	Short	< 50 steps	Unit quizzes; short MOOCs
	Medium	50–200 steps	One term of practice in K–12 math
	Long	> 200 steps	Year-long drilling; daily mobile practice
Structure availability (concept–item relations)	None / Implicit	No reliable item→skill mapping	Only item IDs; unlabeled latent skills
	Explicit static	Fixed, externally provided mapping	Q-matrix/skill tags; curated prerequisite map
	Explicit dynamic	Mapping evolves over time/sessions	Session-wise re-tagging; dynamic knowledge graphs
Temporal irregularity / forgetting cues	Low	Regular intervals; minimal gaps	Daily homework; fixed schedules
	Medium	Moderate variability in gaps	Weekly assignments with occasional delays
	High	Large/irregular gaps; spacing effects salient	Self-paced apps; spaced-repetition platforms
Modality	Binary correctness only	Responses are 0/1 with minimal text	Standard MCQ logs without item text
	Text/code/dialogue modeled	Rich content signals are encoded	Item stems/solutions; source code; tutor–student dialogue
Cohort heterogeneity	Low	Homogeneous population/curriculum	Single grade & course at one school
	Medium	Some curricular/ability diversity	Multiple teachers/courses; mixed ability
	High	Diverse ages/curricula/languages	Cross-age (K–12 + higher ed); multilingual platforms
Cold-start / unseen items	Low	Few new items/students; IID splits	Stable item bank; repeated tests
	Medium	Periodic new items or new cohorts	New students each term; occasional item additions
	High	Frequent unseen items/students; OOD	Inductive/unseen-item splits; cross-course transfer
Operational constraints	Interpretability	Human-readable parameters/explanations required	Coefficients; difficulty/mastery reports
	Robustness / Calibration	Reliability under noise; well-calibrated probabilities	Auto-grading noise; partial credit; ECE targets
	Resource / Privacy	Compute, latency, or data-sharing limits	On-device inference; federated training; PI restrictions

3.4 Harmonizing Outcomes Across Heterogeneous Metrics

Cross-paper synthesis is complicated by heterogeneous metrics (e.g., AUC-ROC, log loss, accuracy, F1, κ) and non-comparable evaluation protocols. To enable aggregation, we produce a two-part summary for each model family within each context (Section 3.3): (1) a weighted win-rate, and (2) a weighted median of normalized ranks.

Step 1: Define comparable instances. For every paper–dataset comparison, the primary metric designated by the authors (AUC preferred when unstated) is taken, yielding an instance. Models within the instance are ranked, where rank 1 indicates the best, and the rank for family f in paper p with n_p total models is normalized using Equation 1. As such, 0 indicates the top performer within that paper–dataset comparison.

$$r_{p,f} = \frac{\text{rank}_{p,f} - 1}{n_p - 1} \tag{1}$$

For example, consider a paper that compares six KT models on a dataset with context c, yielding the following ranking: AKT-R, AKT-NR, DKT, DKVMN, DKT+, SAKT. The normalized rank for AKT-R is $r_{\text{AKT-R}} = \frac{1-1}{6-1} = 0$, and for AKT-NR it is $r_{\text{AKT-NR}} = \frac{2-1}{6-1} = 0.2$. These r values are stored per paper–dataset instance for each family represented.

To limit over-representation from prolific corpora, instances are grouped by (dataset $\times$ family $\times$ context) and capped at $k = 3$ per group, retaining entries via a deterministic quality ordering: protocol quality, reporting completeness, coverage, recency/venue, and reproducibility.

Step 2: Assign quality weights. Each retained paper–dataset instance i was assigned a weight $a > 0$ to reflect evidence quality and risk of bias. These risks have been well documented in previous literature (e.g., [25, 26]). A base weight of 1.0 was multiplied by the following adjustment factors:

- **Protocol quality.** Student-wise chronological and/or out-of-distribution (unseen-student/item) splits: $\times 1.25$. Interaction-wise random or otherwise leakage-prone protocols (e.g., mixing a learner's history across train/test, or including question ID as a feature in both training and test sets): $\times 0.50$.
- **Reporting completeness.** Exact AUC/log loss reported with variance or statistical tests: $\times 1.10$. Directional reporting only (e.g., "outperforms by $\sim$1–3%" with no exact values): $\times 0.75$.

For example, the weight for an instance i that used a student-wise chronological split and reported exact AUC with confidence intervals would be: $a_i = 1.0 \times 1.25 \times 1.10 = 1.375$

Step 3: Compute the weighted win-rate. For each context c and model family f, we define a tie-aware win indicator $w_{i,f} \in [0, 1]$ for each instance i, where $w_{i,f} = 1$ if family f is the sole winner, $w_{i,f} = 0.5$ in the case of a tie between two families, and so on. The weighted win-rate for family f in context c is then computed as:

$$w_{c,f} = \sum_{i \in c} a_i \cdot w_{i,f}$$

where a_i is the quality weight assigned to instance i (see Step 2), and the weights a_i are normalized such that $\sum_{i \in c} a_i = 1$. By construction, $w_{c,f} \in [0, 1]$. Intuitively, $w_{c,f}$ represents the quality-adjusted proportion of wins for model family f within context c.

Step 4: Compute the weighted median of ranks. As a complementary summary, the set $\{(r_{p,f}, \alpha_i)\}_{i \in \mathcal{I}_c}$ is aggregated to a weighted median $\tilde{r}_{c,f}$, providing a robust central tendency of family performance relative to competitors within papers.

Step 5: Sensitivity and bias control. We ran three checks to assess the robustness of the synthesis:

- **Metric sensitivity.** We recomputed all aggregates using only AUC-ROC (discarding papers without AUC) to ensure findings were not artifacts of metric mixing.
- **Protocol sensitivity.** We excluded all higher-risk evaluations (e.g., leakage-prone protocols as mentioned in Step 2) to assess whether rankings remained stable under stricter inclusion criteria.
- **Family granularity.** We compared two grouping strategies: (1) collapsing closely related variants (e.g., SAKT, AKT, SAINT) into broader categories such as attention/Transformer, and (2) treating each variant as a distinct family.

3.5 Reproducibility and Artifacts

All extracted fields and computed labels are stored in a shared spreadsheet (124 rows). The enrichment step—including operational summaries, data context, performance text, and hyperlinks—was scripted in Python using `pandas`, with deterministic de-duplication based on title and URL, and explicit provenance columns for traceability.The context coding scheme and aggregation scripts are available alongside the dataset to support replication, re-weighting, or future extension (e.g., adding new models from 2025–2026).

4 Results

4.1 Corpus Overview

The final corpus comprises 124 KT models/variants spanning probabilistic (e.g., BKT and individualized BKT), generalized logistic and factorization (AFM/PFA/LKT/KTM), deep sequence (DKT and regularized/auxiliary variants), attention/transformer families (SAKT/AKT/SAINT and length-generalization extensions), memory-augmented architectures (DKVMN and successors), graph/heterogeneous models (dual graphs, dynamic graphs, meta-path), time-sensitive models (spacing/forgetting and Hawkes-process variants), contrastive/self-supervised approaches, mixture-of-experts/personalization, uncertainty/robustness-aware methods, and LLM/semantic KT for text/code/dialogue.

Datasets most frequently used include ASSISTments (2009/2012/2015/2017), KDD Cup 2010, Statics, EdNet, and a growing set of programming and dialogue corpora. Metrics are heterogeneous (primarily AUC-ROC; also log loss, accuracy, F1, κ), and split protocols vary (student-wise vs. interaction-wise; chronological vs. random), underscoring the need for the quality adjustments described in Methods.

4.2 Which Models Work Best in Which Contexts

Family-level performance by context is summarized using quality-weighted win rates and weighted median normalized ranks. Representative models and datasets are highlighted to show where each family most often achieves top or near-top performance. Overall, we observe: attention-based models excel on large or long logs; graph-based models perform well when structure is reliable; time-aware models succeed under irregular spacing; and semantic/LLM-based models thrive on text, code, or dialogue data. No universal winner emerges—performance depends on aligning model inductive bias with context.

4.2.1 Large-Scale Logs with Long Histories

On very large, dense logs with long interaction histories, attention/transformer KT tends to lead. In particular, SAINT/SAINT+—which processes item and response streams separately and enriches them with elapsed/lag time—reliably perform well on EdNet ($\approx$131M interactions, 784K learners), with SAINT+ reporting state-of-the-art AUC gains over SAINT on that corpus [12]. Context-aware models such as AKT (Rasch-regularized concept/question embeddings with distance-aware attention) also report consistent AUC improvements across common KT benchmarks (e.g., ASSISTments, Statics) [11]. SAKT's query-conditioned sparse attention likewise shows average AUC gains across multiple datasets [10].

4.2.2 Reliable Concept–Item Structure or Rich Relations

When a rich concept–item structure is available (e.g., stable Q-matrices, high-quality skill graphs), graph-based KT can be especially effective. GKT introduced GNN propagation of student proficiency over a concept graph, and subsequent variants like GIKT incorporate higher-order question–skill relations to improve AUC on several benchmarks [27, 28]. These models work well when concept–item relations are informative and stable (e.g., curated mathematics skill maps such as ASSISTments) and sequences are long enough for graph signals to matter.

4.2.3 Irregular Time Gaps, Recency, and Spacing/Forgetting Effects

In settings where temporal irregularity and forgetting are salient features of the data (e.g., spaced practice logs, long gaps between sessions), models that explicitly encode decay or continuous-time effects tend to lead. DAS3H models per-skill memory decay and multi-skill tagging; HawkesKT uses point-process excitation to capture cross-temporal effects; "DKT-Forget" variants and LPKT incorporate decay or process-consistent learning cells. Empirically, these families improve predictive metrics over RNN baselines on benchmarks with pronounced timing signals [29, 30, 31, 32].

4.2.4 Text, Code, or Dialogue as First-Class Signals

When responses include rich content beyond correct/incorrect (e.g., code, free-text, dialogue), content-aware KT is preferred. Code-DKT uses attention over code features and outperforms BKT/DKT on university programming assignments; Open-Ended KT (OKT) predicts future open-ended responses rather than just correctness in CS education; LLMKT labels skills and correctness in tutor–student dialogues and then traces knowledge, outperforming standard KT on dialogue datasets [33, 14, 34].

4.2.5 Cold-Start and Unseen Items

Under sparsity, cold-start, or strong heterogeneity—typical of platforms with many items/skills but few observations per cell—logistic/factorization families with side features remain highly competitive. Knowledge Tracing Machines (KTM) unify PFA/AFM/mIRT within factorization machines and report superior or comparable AUC on multiple medium-scale datasets (and are robust when observations are sparse or multi-skill) [35]. Question-centric deep models also help when each item has enough data: qDKT shows that replacing skills with items can improve AUC on ASSISTments 2017 (0.72 $\rightarrow$ 0.74 with plain DKT), whereas it overfits on ASSISTments 2009 due to few observations per item—highlighting a context boundary [36].

4.2.6 Heterogeneous Cohorts and Personalization Needs

In cohorts spanning multiple curricula, ages, and study strategies—with mixed ability profiles and long-tail behaviors—mixture-of-experts (MoE) architectures have most frequently led; person-wise routing in RouterKT reports consistent AUC gains across diverse public benchmarks, and option-weighting in WEKT further adapts expert contributions to learner response patterns, with improvements documented on multiple-choice platforms and large, diverse logs [37, 38].

When per-learner data are thinner yet personalization is required, individualized BKT and related hierarchical Bayesian extensions provide competitive performance with interpretable, student-specific parameters via shrinkage across learners [39, 40].

As an intermediate strategy, dynamic student clustering (e.g., DKT-DSC) segments learners by evolving ability and feeds cluster signals to a sequential model, improving prediction under heterogeneous cohorts often observed in datasets like ASSISTments and EdNet [41, 42].

4.2.7 Data-Constrained Settings and Interpretability Requirements

In small-to-medium logs or deployments requiring transparent models, generalized logistic and factorization approaches often perform best. LKT consolidates learner-model features into a constrained logistic framework, achieving strong accuracy and interpretable coefficients across six datasets [43]. KTM extends AFM/PFA/IRT within a factorization machine, handling sparse and multi-skill inputs with competitive performance and fast training [35]. For concept discovery and explainability, SPARFA-Trace jointly models learner knowledge and latent concepts via sparse factor analysis [44, 45]. Classical models like BKT and PFA remain credible in low-data settings and policy-facing applications due to their interpretable parameters [1, 7]. Together, these models offer dependable accuracy with low operational complexity when data or resources are limited.

4.2.8 Noisy Labels, Calibration, and Stability

In settings with auto-grading noise, partial credit, or evaluation volatility, uncertainty- and robustness-aware KT families most frequently led. UKT represents interactions as stochastic distributions and uses Wasserstein self-attention, improving reliability and calibration across multiple public datasets [46]. DTransformer introduces a diagnostic training paradigm that stabilizes predictions across splits while maintaining competitive accuracy on common benchmarks (e.g., ASSISTments, EdNet) [47]. To mitigate shortcutting from raw item identifiers, QDCKT replaces question IDs with difficulty-consistent signals and reports better out-of-distribution generalization under unseen-item protocols [48]. As a complementary strategy, contrastive/self-supervised pretraining (e.g., CL4KT) reduce noise in sequence representations and improves robustness under noisy logs across multiple datasets [49].

5 Discussion

5.1 Main Findings

Across 124 models/variants, no universal winner emerged; rather, performance depended on matching inductive bias to context. Attention/Transformer KT most often led on large, long-history logs, consistent with advantages in modeling long-range dependencies. Graph and dynamic-graph KT were strongest when a reliable concept–item structure existed (static Q-matrices or evolving graphs). Time-sensitive/forgetting-aware families outperformed alternatives under irregular spacing and salient forgetting. LLM/semantic and content-aware KT dominated when text/code/dialogue carried signal, particularly for unseen-item generalization. Mixture-of-experts improved prediction in heterogeneous cohorts. In data-constrained or interpretability-constrained deployments, generalized logistic/factorization (AFM/PFA/LKT/KTM) and psychometric hybrids (e.g., Deep-IRT) delivered competitive accuracy with transparent parameters. Quality-aware weighting and dataset caps reduced optimism from leakage-prone or item-ID–dominated protocols; conclusions were robust in sensitivity analyses.

5.2 Limitations and Future Works

The findings are based on a synthesis of reported results rather than re-implementing on benchmark datasets. As a result, several issues may exist, including metric heterogeneity, incomplete variance reporting, and publication bias. While we addressed these concerns through harmonization, quality-aware weighting, and by capping contributions per dataset to limit over-representation, these steps do not fully eliminate the risks or resolve the issues. Future work should explore alternative weighting strategies and examine how different protocol filters or dataset caps may influence the conclusions.

Additionally, the aggregation prioritizes within-paper comparative evidence. As a result, papers that proposed a model without baselines—or lacked comparable metrics—were excluded from the quantitative aggregation (e.g., rank- or win-rate summaries), though they were still cataloged. This design choice helps prevent misleading cross-paper comparisons based on non-comparable evaluations. However, it may also under-represent emerging model families that have not yet been directly compared to other approaches in head-to-head studies.

Lastly, the present synthesis focuses on the predictive performance of KT models, ranking them based on their effectiveness across various contexts. However, for practical deployment, additional dimensions—such as fairness, computational and data-related costs, and interpretability for teachers and platform developers—are equally important. Future work should investigate how to evaluate and recommend KT models along additional dimensions.

5.3 Conclusion

The evidence synthesized in this study indicates that the relative performance of knowledge tracing (KT) models is context-dependent rather than universally consistent across datasets. No single family of models outperforms others in all settings; instead, the effectiveness of a model is conditioned by the properties of the data and the deployment environment.

From the aggregated literature, several consistent patterns emerge. Attention-based and Transformer models tend to perform well on large datasets with long sequences, while graph-based approaches are more effective when reliable concept–item structures are available. Time-aware models provide advantages under irregular spacing or forgetting dynamics, and semantic or LLM-augmented approaches are most useful when data include rich textual or multi-modal content. Mixture-of-experts approaches support heterogeneous cohorts, and logistic or factorization methods remain strong candidates in smaller datasets or when interpretability is essential.

Beyond the performance of specific model families, this synthesis also highlights the influence of evaluation practices. Variation in data splits, metric reporting, and controls for potential biases (such as item-ID leakage) can alter reported outcomes and complicate cross-paper comparisons. The use of standardized, quality-aware evaluation protocols is therefore critical to ensure results that are both accurate and reproducible.

In conclusion, this work emphasizes that the most appropriate KT model is determined by context. Aligning model assumptions with dataset characteristics, while adopting transparent and standardized evaluation practices, is necessary to advance toward more reliable and actionable applications of KT in real learning environments.

References

[1] 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.

[2] Kurt VanLehn. The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, 16(3):227–265, 2006.

[3] John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. Cognitive tutors: Lessons learned. In *The Journal of the Learning Sciences*, volume 4, pages 167–207, 1995.

[4] Stephen E. Fancsali, Kenneth Holstein, Megan Sandbothe, Steven Ritter, Bruce M. McLaren, and Vincent Aleven. Towards practical detection of unproductive struggle. In *International Conference on Artificial Intelligence in Education*, pages 92–97, Cham, June 2020. Springer International Publishing.

[5] Chao Zhang, Yu Huang, Jun Wang, Di Lu, Wei Fang, John Stamper, and Vincent Aleven. Early detection of wheel spinning: Comparison across tutors, models, features, and operationalizations. In *Proceedings of the International Conference on Educational Data Mining (EDM)*. International Educational Data Mining Society, 2019.

[6] Hao Cen, Kenneth Koedinger, and Brian Junker. Learning factors analysis—a general method for cognitive model evaluation and improvement. In *Intelligent Tutoring Systems*, pages 164–175. Springer, 2006.

[7] Philip I Pavlik, Hao Cen, and Kenneth R Koedinger. Performance factors analysis–a new alternative to knowledge tracing. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, pages 531–538. IOS Press, 2009.

[8] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In *Advances in neural information processing systems*, volume 28, 2015.

[9] Jian 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*, pages 765–774, 2017.

[10] Shalini Pandey and George Karypis. A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837*, 2019.

[11] 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*, pages 2330–2339, 2020.

[12] Dongmin Shin, Jaewook Shim, Youngduck Choi, and Jungwoo Kim. Saint+: Integrating temporal features for ednet correctness prediction. In *Proceedings of the 14th International Conference on Educational Data Mining*, 2021.

[13] Hiromi Nakagawa, Yusuke Iwasawa, and Yutaka Matsuo. Graph-based knowledge tracing: Modeling student proficiency using graph neural network. In *Proceedings of the 12th International Conference on Educational Data Mining*, 2019.

[14] Naiming Liu, Zichao Wang, Richard Baraniuk, and Andrew Lan. Open-ended knowledge tracing for computer science education. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

[15] Alexandra Scarlatos, Ryan S Baker, and Andrew Lan. Exploring knowledge tracing in tutor-student dialogues using llms. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, 2025.

[16] Joseph Beck, Korinn Ostrow, and Yan Wang. Students vs. skills: Partitioning variance explained in learner models. *Student modeling from different aspects*, pages 2–9, 2016.

[17] Mingyu Feng, Neil T Heffernan, and Kenneth R Koedinger. Addressing the assessment challenge with an online system that tutors as it assesses. In *User Modeling and User-Adapted Interaction*, volume 19, pages 243–266, 2009.

[18] Leena Razzaq, Mingyu Feng, Goss Nuzzo-Jones, Neil T Heffernan, KR Koedinger, Brian Junker, Steven Ritter, Andrea Knight, C Aniszczyk, S Choksey, et al. The assistment project: Blending assessment and assisting. In *Proceedings of the 12th annual conference on artificial intelligence in education*, number 0231773, 2005.

[19] Kenneth R Koedinger, Ryan SJd Baker, Kyle Cunningham, Alida Skogsholm, Brett Leber, and John Stamper. A data repository for the edm community: The pslc datashop. *Handbook of educational data mining*, 43:43–56, 2010.

[20] Youngduck Choi, Youngnam Lee, Jaewe Heo Cho, Kyungmin Park, Yeon-Ju Cha, Dongmin Shin, Jongseong Kim, Jungwoo Kim, and Jung-Woo Ha. Ednet: A large-scale hierarchical dataset in education. In *Proceedings of the 29th International Conference on World Wide Web*, 2020.

[21] Haw-Shiuan Chang, Hwai-Jung Hsu, Kuan-Ta Chen, et al. Modeling exercise relationships in e-learning: A unified approach. In *EDM*, pages 532–535, 2015.

[22] Burr Settles, Chris Brust, Erin Gustafson, Masato Hagiwara, and Nitin Madnani. Second language acquisition modeling. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 56–65, 2018.

[23] Thomas Gervet, Kenneth R Koedinger, Jonas Schneider, and Tom Mitchell. When is deep learning the best approach to knowledge tracing? *Journal of Educational Data Mining*, 12(3):31–54, 2020.

[24] Robin Schmucker, Jingbo Wang, Shijia Hu, and Tom M Mitchell. Assessing the performance of online students–new data, new approaches, improved accuracy. *arXiv preprint arXiv:2109.01753*, 2021.

[25] Yahya Badran and Christine Preisach. Enhancing knowledge tracing through leakage-free and recency-aware embeddings. *arXiv preprint arXiv:2508.17092*, 2025.

[26] Zachary A Pardos, Sujith M Gowda, Ryan SJd Baker, and Neil T Heffernan. The sum is greater than the parts: Ensembling models of student knowledge in educational software. *ACM SIGKDD explorations newsletter*, 13(2):37–44, 2012.

[27] Hiromi Nakagawa, Yusuke Iwasawa, and Yutaka Matsuo. Graph-based knowledge tracing: Modeling student proficiency using graph neural networks. In *International Conference on Learning Representations (ICLR) Workshop*, 2019. Presented at ICLR 2019.

[28] Yang Yang, Jian Shen, Yanru Qu, Yunfei Liu, Kerong Wang, Yaoming Zhu, Weinan Zhang, and Yong Yu. Gikt: A graph-based interaction model for knowledge tracing. *arXiv*, arXiv:2009.05991, 2020.

[29] B. Choffin, F. Popineau, Y. Bourda, and J.-J. Vie. DAS3H: Modeling student learning and forgetting for optimally scheduling distributed practice of skills. In *Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019)*, pages 29–39, 2019.

[30] Shuanghong Shen, Qi Liu, Enhong Chen, Zhenya Huang, Wei Huang, Yu Yin, Yu Su, and Shijin Wang. Learning process-consistent knowledge tracing (lpkt). In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21)*, pages 1452–1460, 2021. doi: 10.1145/3447548.3467237.

[31] Koki Nagatani, Qian Zhang, Masahiro Sato, Yan-Ying Chen, Francine Chen, and Tomoko Ohkuma. Augmenting knowledge tracing by considering forgetting behavior. In *Proceedings of the World Wide Web Conference (WWW '19)*, pages 3101–3107, 2019. doi: 10.1145/3308558.3313565.

[32] Shuanghong Shen, Qi Liu, Enhong Chen, Zhenya Huang, Wei Huang, Yu Yin, Yu Su, and Shijin Wang. Learning process-consistent knowledge tracing. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 1452–1460, 2021.

[33] Yang Shi, Min Chi, Tiffany Barnes, and Thomas W. Price. Code-dkt: A code-based knowledge tracing model for programming tasks. In *Proceedings of the 15th International Conference on Educational Data Mining (EDM 2022)*, 2022.

[34] Alexander Scarlatos, Ryan S. Baker, and Andrew Lan. Exploring knowledge tracing in tutor-student dialogues using llms. In *Proceedings of the 15th Learning Analytics and Knowledge Conference (LAK 2025), Dublin, Ireland*. ACM, 2025.

[35] Jill-Jênn Vie and Hisashi Kashima. Knowledge tracing machines: Factorization machines for knowledge tracing. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI 2019)*, pages 750–757, 2019. Presented at AAAI 2019; preprint available at `https://arxiv.org/abs/1811.03388`.

[36] Shashank Sonkar, Andrew E. Waters, Andrew S. Lan, Phillip J. Grimaldi, and Richard G. Baraniuk. qdkt: Question-centric deep knowledge tracing. In *Proceedings of the 13th International Conference on Educational Data Mining (EDM 2020)*, pages 677–681, 2020. preprint also available on arXiv:2005.12442.

[37] Han Liao and Shuaishuai Zu. Routerkt: Mixture-of-experts for knowledge tracing. *arXiv preprint arXiv:2504.08989*, 2025.

[38] Tao Huang, Xinjia Ou, Huali Yang, Shengze Hu, Jing Geng, Zhuoran Xu, and Zongkai Yang. Pull together: Option-weighting-enhanced mixture-of-experts knowledge tracing. *Expert Systems with Applications*, 248: 123419, 2024.

[39] Michael V. Yudelson, Kenneth R. Koedinger, and Geoffrey J. Gordon. Individualized bayesian knowledge tracing models. In *Proceedings of the 16th International Conference on Artificial Intelligence in Education (AIED 2013)*, volume 7926 of *Lecture Notes in Computer Science*, pages 171–180. Springer, 2013.

[40] Michael V Yudelson. Individualizing bayesian knowledge tracing. are skill parameters more important than student parameters?. *International Educational Data Mining Society*, 2016.

[41] Sein Minn, Yi Yu, Michel C. Desmarais, Feida Zhu, and Jill Jenn Vie. Deep knowledge tracing and dynamic student classification for knowledge tracing. In *Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM)*, pages 1182–1187. IEEE, 2018.

[42] S. Minn, F. Zhu, and M. C. Desmarais. Improving knowledge tracing model by integrating problem difficulty. *Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1505–1506, 2018.

[43] Philip I. Jr. Pavlik, Luke G. Eglington, and Leigh M. Harrell-Williams. Logistic knowledge tracing: A constrained framework for learner modeling. *IEEE Transactions on Learning Technologies*, 14(5):624–639, 2021. doi: 10.1109/TLT.2021.3128569.

[44] Andrew S Lan, Christoph Studer, and Richard G Baraniuk. Time-varying learning and content analytics via sparse factor analysis. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 452–461, 2014.

[45] Studer Christoph Lan, Andrew S and Richard G Baraniuk. Quantized matrix completion for personalized learning. *arXiv preprint arXiv:1412.5968*, 2014.

[46] Weihua Cheng, Hanwen Du, Chunxiao Li, Ersheng Ni, Liangdi Tan, Tianqi Xu, and Yongxin Ni. Uncertainty-aware knowledge tracing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27905–27913, 2025.

[47] 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 (WWW '23)*, pages 855–864, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394161. doi: 10.1145/3543507.3583255.

[48] Guimei Liu, Huijing Zhan, and Jung-jae Kim. Question difficulty consistent knowledge tracing. In *Proceedings of the ACM Web Conference 2024*, pages 4239–4248, 2024.

[49] Wonsung Lee, Jaeyoon Chun, Youngmin Lee, Kyoungsoo Park, and Sungrae Park. Contrastive learning for knowledge tracing. In *Proceedings of the ACM Web Conference 2022 (WWW '22)*, pages 2330–2338. Association for Computing Machinery, 2022. doi: 10.1145/3485447.3512105.

Agents4Science AI Involvement Checklist

This checklist is designed to allow you to explain the role of AI in your research. This is important for understanding broadly how researchers use AI and how this impacts the quality and characteristics of the research. **Do not remove the checklist! Papers not including the checklist will be desk rejected.** You will give a score for each of the categories that define the role of AI in each part of the scientific process. The scores are as follows:

- **[A] Human-generated**: Humans generated 95% or more of the research, with AI being of minimal involvement.
- **[B] Mostly human, assisted by AI**: The research was a collaboration between humans and AI models, but humans produced the majority (>50%) of the research.
- **[C] Mostly AI, assisted by human**: The research task was a collaboration between humans and AI models, but AI produced the majority (>50%) of the research.
- **[D] AI-generated**: AI performed over 95% of the research. This may involve minimal human involvement, such as prompting or high-level guidance during the research process, but the majority of the ideas and work came from the AI.

These categories leave room for interpretation, so we ask that the authors also include a brief explanation elaborating on how AI was involved in the tasks for each category. Please keep your explanation to less than 150 words.

1. **Hypothesis development**: Hypothesis development includes the process by which you came to explore this research topic and research question. This can involve the background research performed by either researchers or by AI. This can also involve whether the idea was proposed by researchers or by AI.

   Answer: **[A]**

   Explanation: The research question(namely, what knowledge tracing models work best in what contexts) was proposed by human researchers with domain expertise in the field. These researchers initially identified the gap in the literature and formulated the research idea. As such, in the current work, AI was not prompted to generate or revise the research question; rather, it was tasked with assisting as a research partner by performing a systematic review, gathering data, and helping to derive conclusions.

2. **Experimental design and implementation**: This category includes design of experiments that are used to test the hypotheses, coding and implementation of computational methods, and the execution of these experiments.

   Answer: **[C]**

   Explanation: To address the research question and the impracticality of conducting an empirical analysis, as noted in the article, the human researchers decide to adopt a systematic review approach for the current study. To ensure rigor and transparency, we designed a three-step prompt: (1) collect as many relevant papers as possible on the topic, (2) extract and synthesize key information to answer the research question, and (3) compare models across different contexts. While the overall approach was proposed by human researchers, AI contributed specific implementation details. In particular, it suggested (1) inclusion and exclusion criteria, (2) a contextual operationalization of datasets using eight dimensions, along with coding for each dataset, and (3) computational methods, such as weighted ranking and win-rate, for comparing model performance.

3. **Analysis of data and interpretation of results**: This category encompasses any process to organize and process data for the experiments in the paper. It also includes interpretations of the results of the study.

   Answer: **[D]**

   Explanation: This study relied heavily on AI for data collection, analysis, and interpretation, given the large volume of papers and data involved. At each stage of the three-step prompt, AI was given a specific task and engaged with iteratively until results meeting the intended objective were obtained(even if done in a fashion very different than how the human researchers would have done it). For example, during the data collection phase, human researchers prompted the AI multiple times to refine and expand the set of retrieved papers. Additionally, the AI was asked to explain and justify its methodological choices throughout the process. In many cases, the AI's approach was unusual for the field and different than what was expected by the authors, such as the unorthodox approach to paper weighting adopted

4. **Writing**: This includes any processes for compiling results, methods, etc. into the final paper form. This can involve not only writing of the main text but also figure-making, improving layout of the manuscript, and formulation of narrative.

Answer: **[C]**

Explanation: At the end of the analysis, AI was prompted to draft the Methods and Results sections. Human researchers then reviewed and edited the drafts to enhance readability and elaborate on underdeveloped points by adding examples and clarifications. The first step for major edits was to ask the AI to rewrite to clarify or address a point. For the remaining sections, AI was provided with an outline to generate initial drafts, which were subsequently refined by human researchers.

5. **Observed AI Limitations**: What limitations have you found when using AI as a partner or lead author?

   Description: **Writing:** One key limitation is over-simplification in writing. The AI often struggled to craft coherent narratives that provide sufficient context for human readers. It tends to present information in a fragmented or surface-level way, requiring frequent prompting to unpack ideas or explain concepts more thoroughly. **Conducting Scientific Research:** While AI is fairly strong at suggesting methodological approaches, some of its recommendations can be arbitrary or lack empirical justification. For instance, in the current study, it proposed novel and seemingly arbitrary paper weighting schemes, which were creative but not grounded in prior evidence or validation, and which likely would have attracted negative attention from reviewers in the field.

Agents4Science Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims presented in the abstract and introduction accurately reflect the paper's contributions and scope, clearly outlining the research question addressed and the key findings.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are explicitly discussed in the Limitations and Future Work subsection. The paper identifies three main constraints that motivate future research: (1) potential biases introduced by metric heterogeneity, with alternative mitigation approaches left for future work; (2) exclusion of models from papers without within-study comparisons, which limits coverage of some emerging families; and (3) an emphasis on predictive performance, with less attention to other critical deployment factors such as fairness, computational and data costs, and interpretability. In addition, potential concerns about the approach to weighting papers were discussed.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper does not present formal theorems or mathematical proofs, but all methodological assumptions underlying the synthesis are made explicit. The procedures for corpus construction, metric harmonization, rank normalization, and quality/bias weighting are fully documented. Supplementary materials include (1) a spreadsheet cataloging all included papers, (2) Python scripts implementing the ranking and win-rate aggregation, and (3) a summary spreadsheet presenting the aggregated results. Together, these resources provide a complete and transparent account of the assumptions and analyses needed to reproduce the findings.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The analysis is documented in the Methods section, including the criteria and decisions used throughout the process. While the stochastic nature of LLMs may lead to slight variations in the set of articles retrieved upon replication, we do not expect these differences to meaningfully impact the overall results or conclusions of the study.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We plan to upload the data and intermediary files to a public repository with open access.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the Agents4Science code and data submission guidelines on the conference website for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [NA]

   Justification: The paper does not present new experiments; rather, it is a systematic literature synthesis that aggregates and harmonizes reported results from prior studies. However, all details are provided including corpus selection, coding of dataset contexts, harmonization of evaluation metrics, and quality/bias adjustments, to facilitate understanding of the results.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: The paper does not present new experiments or statistical significance tests and therefore does not report error bars or confidence intervals for the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, or overall run with given experimental conditions).

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [NA]

Justification: The paper does not involve running new model training or large-scale experiments, but instead synthesizes reported results from prior studies. As such, compute resources such as GPU/CPU type, memory, or execution time are not relevant. The only analyses performed were data processing and aggregation using spreadsheets and Python scripts, which require minimal computational resources and are reproducible on a standard personal computer. In total, we estimate that roughly 200 calls were made to produce this work—about 110–135 for conducting the research and 65–85 for writing the manuscript.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the Agents4Science Code of Ethics (see conference website)?

Answer: [Yes]

Justification: The research fully conforms with the Agents4Science Code of Ethics. The study does not involve human subjects, sensitive data, or interventions; it is a systematic synthesis of published literature on knowledge tracing. All data used are publicly available from prior publications or benchmark repositories.

Guidelines:

- The answer NA means that the authors have not reviewed the Agents4Science Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper explicitly discusses broader impacts. On the positive side, the synthesis provides actionable guidance on selecting knowledge tracing models that are better suited to specific contexts. On the negative side, the paper notes risks such as over-reliance on predictive performance at the expense of interpretability.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations, privacy considerations, and security considerations.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies.