Deep Knowledge Tracing architectures: a technical comparison

Deep Knowledge Tracing models have evolved from simple LSTMs to sophisticated transformer architectures, with **AKT consistently achieving the highest accuracy** (AUC 0.83-0.84 on ASSISTments 2009) umass while simpler models like DKT remain competitive due to the short-term dependency nature of educational data. The choice of architecture depends heavily on dataset scale, computational constraints, and interpretability requirements—transformer-based models excel on large datasets like EdNet (**784K students, 95M+ interactions**) Amazonaws +3 but may underperform on smaller benchmarks where LSTM-based DKT surprisingly outperforms attention-based SAKT and SAINT. umass

Architecture fundamentals reveal different design philosophies

The evolution from DKT to modern transformers reflects fundamentally different approaches to modeling student knowledge. **DKT (Piech et al., 2015)** uses LSTMs to maintain a latent knowledge state through recurrent hidden states, ACM Computing Surveys +3 processing interactions sequentially ResearchGate with $O(n \cdot d^2)$ complexity. Each interaction is encoded as a tuple (exercise, correctness), typically one-hot encoded into a 2M-dimensional vector (M = number of exercises) PubMed Central or compressed via random projections for large exercise spaces. stanford

Model	Architecture	Complexity	Parameters	Key Innovation
DKT	LSTM (1-2 layers)	$O(n \cdot d^2)$	100K-500K	First DL approach to KT
DKT+	LSTM + regularization	$O(n \cdot d^2)$ + overhead	100K-500K	Reconstruction + waviness loss
SAKT	Single attention block	$O(n^2 \cdot d)$	500K-2M	Exercise queries, interaction keys/values
SAINT	Encoder-decoder transformer	$O(N \cdot n^2 \cdot d)$	1M-5M	Separated exercise/response streams
AKT	Context-aware attention	$O(n^2 \cdot d)$	1M-3.3M	Rasch embeddings + exponential decay

SAKT (Pandey & Karypis, 2019) introduced attention to knowledge tracing umass using a unique Query-Key-Value formulation: exercises serve as queries while past interaction embeddings (exercise + correctness × E) provide keys and values. ed This enables the model to identify which historical exercises are most relevant to the current prediction. SAKT uses **5 attention heads** with learnable positional embeddings, umass achieving **17-46× faster training** than DKT through parallelization despite quadratic sequence complexity.

SAINT's encoder-decoder architecture processes exercise sequences through the encoder while the decoder handles response sequences with cross-attention. ACM Digital Library This separation—verified empirically—allows deeper attention stacking Rtest Vertexdoc (typically 4 layers, **d=512**) and achieves state-of-the-art performance on EdNet. SAINT+ extends this with continuous elapsed-time embeddings (`v_et = et · w_elapsed`) and categorical lag-time embeddings (150 discrete bins), adding temporal awareness that improves AUC by **+1.25%**. Vertexdoc +2

AKT uniquely incorporates psychometric theory through Rasch model-based embeddings: $x_t = c_{c_t} + \mu_{e_t} \cdot d_{c_t}$, where μ represents a learnable difficulty parameter per question. arxiv Its monotonic attention mechanism applies exponential decay ($\exp(-\theta \cdot d(t,\tau))$) to attention scores, modeling forgetting curves from cognitive science. umass This context-aware distance measure creates "spikes" for concept-relevant historical interactions while down-weighting temporally distant ones.

Implementation requirements vary significantly across model families

Framework dependencies and code availability are well-established across all models. The **pyKT toolkit** (github.com/pykt-team/pykt-toolkit, NeurIPS 2022) provides standardized PyTorch implementations for 15+ models including all architectures discussed here. GitHub github Official repositories include:

- **DKT+**: github.com/ckyeungac/deep-knowledge-tracing-plus (TensorFlow 1.2+) github
- **AKT**: github.com/arghosh/AKT (PyTorch 1.2.0) GitHub umass
- **SAINT implementations**: arshadshk/SAINT-pytorch, Chang-Chia-Chi/SaintPlus-Knowledge-Tracing-Pytorch

Memory footprint scales quadratically with sequence length for attention-based models. For SAINT with L=100 and d=512, expect **2-4 GB GPU memory** during training; extending to L=500 requires **20-30 GB**. DKT's linear sequence complexity keeps memory requirements modest at **2-4 GB** regardless of sequence length. AKT, with approximately **3.3M parameters** (the largest among compared models), requires 4-8 GB for typical configurations. arXiv

Standard training configurations across implementations use batch sizes of **64-2048**, sequence lengths capped at **100-200**, embedding dimensions of **128-512**, and Adam optimizer with learning rate **1e-4 to 1e-3**. The pyKT benchmark recommends: lr=3×10^{-4}, batch_size=64, epochs=100-300 with early stopping, embedding_dim=128.

Training data requirements and benchmark datasets

Minimum dataset sizes vary substantially by architecture complexity:

Model	Min Students	Min Interactions	Cold-Start Threshold
DKT/DKT+	1,000+	50,000+	5-10 interactions
SAKT	5,000+	200,000+	Similar to DKT
SAINT/SAINT+	10,000+	500,000+	Benefits from scale
AKT	2,000+	100,000+	Better with Rasch embeddings

ASSISTments 2009 (346K interactions, 4,217 students, 123 KCs) Educationaldatamining remains the canonical benchmark, GitHub ACM Digital Library though the pyKT team identified significant **label leakage issues** in many published results—expanding multi-KC questions inflates AUC by **8-13%**. Liner ResearchGate **EdNet-KT1** (95M interactions, 784K students) Amazonaws represents the largest public benchmark ACM Other conferences Semantic Scholar and where SAINT/SAINT+ excel, while **Statics2011** (189K interactions, 333 students) ACM Digital Library tests performance on smaller, denser datasets. umass

Data format requirements follow a consistent pattern across models:

```
student_id, question_id, skill_id (optional), correct (0/1), timestamp (optional)
```

Cold-start performance remains challenging across all architectures. Research shows predictions stabilize around **10-20 interactions**, with SAKT showing marginally higher initial accuracy. Recent LLM-based approaches (CLST) demonstrate up to **24.52% improvement** in cold-start scenarios by leveraging semantic understanding of question content.

Real-time inference performance shows GPU advantages at scale

Latency characteristics favor LSTMs for single predictions but GPUs dominate batch processing:

Configuration	DKT (CPU)	SAKT (CPU)	SAINT (V100)	AKT (V100)
Single prediction, L=100	1-5ms	5-15ms	10-20ms	10-20ms
Batch=64, L=100	50-100ms	200-300ms	15-30ms	15-30ms
Batch=64, L=500	100-200ms	1-2s	100-200ms	100-200ms

Production deployments require <100ms latency for interactive tutoring. Riiid's Santa TOEIC platform serves 780K+ users Amazonaws +2 using SAINT+ variants, demonstrating transformer feasibility at scale. Optimization strategies include KV-caching for repeated encoder outputs, ONNX/TensorRT compilation, mixed-precision (FP16) inference reducing memory ~50%, and sequence truncation to most recent interactions.

Training efficiency shows SAKT's parallelization advantage: **1.4 seconds per epoch** on ASSISTments 2009 versus 45 seconds for DKT and 65 seconds for DKT+—a **32-46× speedup**. However, this advantage diminishes with longer sequences where quadratic complexity dominates.

Accuracy benchmarks reveal surprising patterns

The pyKT standardized benchmark (5-fold CV, question-level prediction) NeurIPS produces notably different results than original papers:

Model	AS2009	AS2015	AS2017	EdNet	Statics2011
DKT	0.755	0.702	0.734	~0.76	0.822
DKT+	0.769	0.702	0.740	—	0.822
SAKT	0.727	0.710	0.712	~0.75	0.775
SAINT	0.698	0.689	0.703	~0.78	0.779
AKT	**0.788**	**0.767**	0.730	~0.79	**0.822**
SAINT+	—	—	—	**0.791**	—

A critical finding: **DKT and DKT+ consistently outperform SAKT and SAINT on smaller datasets**, contrary to expectations from NLP where attention mechanisms dominate. This reflects the fundamental difference in knowledge tracing: educational sequences exhibit **strong recency effects** rather than long-range dependencies, making LSTM's sequential bias advantageous. Liner +2 AKT's monotonic attention explicitly models this through exponential decay, explaining its superior performance. umass

The original DKT paper reported **0.86 AUC** on ASSISTments 2009, demonstrating a 25% improvement over BKT (0.69). Stanford University +2 However, subsequent standardized evaluations show more modest improvements, with AKT achieving 0.788—still a significant advancement but highlighting the importance of consistent evaluation protocols.

Interpretability and temporal dynamics differ fundamentally

Attention-based models offer visualization capabilities unavailable in LSTMs. **SAKT's attention weights** reveal which past exercises influence current predictions—the original paper demonstrated perfect clustering of 5 hidden concepts in synthetic data. stanford ed For real questions, attention heaviest on conceptually similar exercises (e.g., "Division Fractions" weighted 0.99 for "Scale Factor" predictions).

AKT provides dual interpretability: attention weights show temporal relevance while Rasch difficulty parameters (μ) quantify question hardness on a continuous scale. arXiv umass The exponential decay parameter θ is learnable per model, allowing different "forgetting rates" across deployments.

DKT+'s contribution to interpretability focuses on **behavioral consistency**. The reconstruction loss ($r = \Sigma_t \ell(y_t^T \cdot \delta(q_t), a_t)$) ensures predictions for skills practiced correctly increase, while waviness regularization ($w = \Sigma_t \|y_{t+1} - y_t\|$) prevents dramatic prediction fluctuations. This improves the consistency metric m1 (proportion of predictions changing correctly) from **0.59 to 0.81**.

Forgetting modeling approaches vary substantially:

- **LPKT**: Explicit forgetting gate based on interval time between interactions

- **AKT**: Exponential decay attention with context-aware distance (umass)
- **DKT-Forget**: Multiple forgetting information features added to input
- **SAINT+**: Lag time embeddings (categorical, 150 bins up to 1440 minutes) (Rtest)

Scalability to large skill spaces and recent advances

Large skill/concept spaces (>1000 KCs) present computational challenges that different architectures address distinctly. **AKT's Rasch embeddings** require only one scalar per question rather than separate embeddings, reducing parameters from 2CD to (C+2)D + Q. (umass) Graph-based approaches like **GKT** explicitly model KC relationships but add O(|E|) edge computations.

XES3G5M (865 KCs, 7,652 questions, 5.5M interactions) represents the largest KC space in standard benchmarks. (NIPS) Production systems like Squirrel AI's nano-level decomposition push further: 30,000 knowledge points for junior high math alone, (Wikipedia) serving 24M+ students across 60,000+ schools. (Wikipedia)

Recent innovations (2023-2025) focus on stability and interpretability over pure accuracy:

- **simpleKT (ICLR 2023)**: Demonstrates standard attention + Rasch embeddings matches sophisticated mechanisms
- **DTransformer (WWW 2023)**: Contrastive learning for stable knowledge state diagnosis
- **CL4KT (WWW 2022)**: Contrastive framework with hard negative mining from answer reversal (ACM Digital Library)
- **GRKT (2024)**: Graph-based approach achieving **1.0 consistency metric**
- **Mamba4KT (2024)**: Efficient sequence modeling via Mamba architecture, though underperforms on large datasets (arXiv)

Conclusion

The knowledge tracing landscape reveals that **architectural sophistication doesn't guarantee superior performance**. AKT's combination of psychometric embeddings and cognitively-motivated attention decay achieves the best accuracy across most benchmarks, (umass) (ResearchGate) but DKT remains surprisingly competitive—particularly on smaller datasets where transformer overhead provides diminishing returns. (Liner) For production deployment, SAINT+ offers the best balance of accuracy on large-scale data (EdNet) and real-time inference capability, while simpleKT demonstrates that proper embeddings matter more than complex attention mechanisms.

Key implementation decisions should consider: (1) dataset scale—use DKT for <100K interactions, AKT or SAINT+ for larger; (2) latency requirements—DKT for single-student real-time, batch processing enables transformers; (3) interpretability needs—AKT provides both attention visualization and difficulty parameters;

(4) cold-start scenarios—all models struggle with <10 interactions, consider LLM-augmented approaches for cold-start-heavy applications. The pyKT toolkit provides standardized implementations and evaluation protocols essential for reproducible benchmarking across any architecture choice. arXiv GitHub