Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Predictive Reliability of Knowledge Tracing Models Under Sparse and Noisy Student Response Data

Meixian Jin

jocelyn910304@gmail.com

Emporia State University

Research Article

Keywords: Knowledge Tracing; Predictive Reliability; Probability Calibration; Sparse Response Histories; Response Noise; Educational Artificial Intelligence

Posted Date: November 19th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11682101/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Predictive Reliability of Knowledge Tracing Models Under Sparse and Noisy Student Response Data

Meixian Jin
Emporia State University
Emporia, Kansas, USA
mjin1@g.emporia.edu

Knowledge tracing (KT) models are usually compared by ranking metrics, yet adaptive systems act on predicted probabilities, which may degrade when the response history is short or noisy. This study audits next-response probabilities of five predictors (a smoothed item-accuracy baseline, Bayesian knowledge tracing with forgetting, 2PL item response theory, deep knowledge tracing (DKT), and a simplified attention-based model), holding the predicted events fixed while varying visible history length (3–50 events), history label flips (0–20%), and guessing and slipping separately. Learners were simulated from two generators with different assumptions. We report ROC-AUC, Brier score, negative log-likelihood, and expected calibration error with learner-level bootstrap intervals, plus high-confidence errors, and compare temperature with Platt scaling. Model ordering depended on the generator and on item information: under the dynamic generator no model beat the item baseline on Brier score at three visible events, and there DKT with item outputs matched the attention-based model. Truncating a long record made every history-based model under-predict (by up to 0.086), because each began the window from a state fitted to earlier practice; when three events were a learner's complete record, the same models were nearly unbiased. Temperature scaling, which has no intercept, left this offset in place in most cases and transferred inconsistently from long histories, whereas Platt scaling fitted in the target condition removed it. Label flips lowered discrimination and, with short histories, mostly erased the advantage over the baseline. The findings are a controlled reference for validation on real interaction logs.

Keywords: Knowledge Tracing; Predictive Reliability; Probability Calibration; Sparse Response Histories; Response Noise; Educational Artificial Intelligence

1. INTRODUCTION

A knowledge tracing (KT) model is rarely asked only to rank learners. In an adaptive practice system its output, the probability that the next response is correct, decides what is reviewed, which exercise comes next, or whether another assessment is requested (Abdelrahman et al., 2022; Pelánek, 2017). Anyone acting on a prediction of 0.90 is relying on it being wrong about one time in ten. Yet most KT evaluation still reports discrimination, usually the area under the ROC curve (AUC), on benchmark collections (Liu et al., 2022; Pelánek, 2015). Discrimination and calibration are different properties. A model can order responses well and still be systematically overconfident or timid, and AUC will not show it (Guo et al., 2017; Niculescu-Mizil and Caruana, 2005).

The models usually compared make different bets. Bayesian knowledge tracing (BKT) treats mastery of a skill as a latent binary state, updated after every response and tied to

answers through guess and slip parameters (Corbett and Anderson, 1995). Item response theory (IRT) assumes a stable ability and fixed item difficulties (Pelánek, 2017). Deep knowledge tracing (DKT) hands the problem to a recurrent network (Piech et al., 2015), and AKT adds attention with a monotonic distance decay and Rasch-type question embeddings (Ghosh et al., 2020). Whether the deep models win depends on the data (Gervet et al., 2020; Khajah et al., 2016), which is a reason to doubt that any single ranking of models travels well.

Two ordinary conditions make the calibration question harder. First, a model often sees only part of a learner's record, for example when earlier records are missing, when logs are truncated, or when the model is applied to a recent window of activity. This differs from cold start, in which a learner has answered only a few items, and the two cases are compared in Section 3.5. Second, responses are a noisy signal: guessing and slipping limit what a few answers can reveal, and they are hard to disentangle from the latent state itself (Baker et al., 2008; Beck and Chang, 2007). Logs may also carry outright wrong labels (Frénay and Verleysen, 2014). Elsewhere in machine learning, predicted probabilities are known to degrade under dataset shift even when ranking holds up (Ovadia et al., 2019). How far that carries over to KT predictors, and to which of them, is not settled.

Uncertainty in KT is not a neglected topic. UKT represents interactions as distributions and adds a contrastive loss aimed at aleatory noise (Cheng et al., 2025). Dynamic LENS keeps epistemic uncertainty through variational encoders and Bayesian state-space updates (Christie et al., 2024), and selective prediction with Monte Carlo dropout lets a model defer when it is least certain (Mitton et al., 2026). Beyond education, uncertainty-aware and anomaly-aware designs have recently been proposed for resource scheduling in LLM serving and in service meshes (Gu et al., 2026; Tao, 2026), where automated decisions likewise depend on reliable probability estimates.

One distinction matters for everything below. Estimating a knowledge state is not the same as predicting the next response. BKT's mastery probability, IRT's ability parameter, and the hidden units of DKT and AKT are different objects, and real logs provide no labelled version of any of them. We therefore score every model on the one quantity they share, the observed next response, and we do not evaluate latent-state recovery. Attention weights and hidden units are not read as reasons for a learner's errors.

Calibration of KT models has been examined before. Gervet et al. (2020) compared logistic-regression models, a BKT variant, and deep models on nine real datasets, inspected reliability diagrams for their best logistic-regression model and DKT, showed that DKT's performance depends on whether skills or items serve as its inputs and outputs, and observed that the relative standing of DKT and logistic regression changed with the number of interactions available per learner. In real logs, however, the visible history, record errors, item heterogeneity, and the position of a predicted response in a learner's sequence vary together, so their separate effects on calibration cannot be read off such comparisons.

We pose three research questions.

RQ1. How does a limited visible response history affect predictive discrimination and probability calibration across knowledge tracing models?

RQ2. How do guessing, slipping, and corrupted historical labels affect high-confidence prediction errors?

RQ3. Can simple post-hoc calibration improve reliability, and when do its benefits fail to transfer across data conditions?

To answer them, the predicted events are held fixed while the visible history length, the history label noise, and the response randomness are varied one at a time, under two

generators with different assumptions, so that no model is tested only on data built from its own assumptions. A calibration map refitted in the target condition is distinguished from one carried over from clean, long histories. Control comparisons then separate model family from the item information a model uses, the training scheme from the matching of training to test conditions, and truncated histories from genuinely short ones. The main new observation concerns truncation. When only the last few events of a long record were visible, every history-based model under-predicted, because it began the window from a state fitted to an earlier stage of practice. For most models the offset grew as the window shrank, and it was largely absent when the same few events were a learner's complete record. A temperature, which has no intercept, left it in place in most cases, and a temperature fitted on long histories, where the offset is small, could not anticipate it; a map with an intercept fitted in the target condition removed it. The paper proposes no new algorithm. Its contributions are the controlled comparison, the mechanism it exposes, and a reproducible protocol whose code and saved predictions rebuild every table and figure (Section 2.6).

2. RESEARCH METHODS

2.1. DATA SOURCES AND SIMULATION DESIGN

Each replicate contains $N = 3000$ simulated learners, each answering $T = 60$ items from a bank of 200 items grouped into 10 skills of 20 items. The skill sequence stays on the previous skill with probability 0.6 and otherwise switches to a random skill, which produces practice runs; the item is drawn uniformly within the skill. Five independent replicates (seeds 1–5) each generate a new item bank and new learners. A separate development seed (9001) is used only for hyperparameter selection. Learners are split 60/15/10/15 into training, validation, calibration, and test sets (1,800/450/300/450).

Synthetic-KT-A is a two-state learning, forgetting, guessing, and slipping process. For learner i and skill k the latent state $z_{ik} \in \{0,1\}$ is updated over time. Write z_{it} for the state of learner i on the skill of the item answered at step t. The response at step t, to item j, follows

$$P(y_{it} = 1 \mid z_{it} = 1) = 1 - s_j, \qquad P(y_{it} = 1 \mid z_{it} = 0) = g_j, \tag{1}$$

where s_j is the probability of an incorrect answer when the skill is mastered and g_j the probability of a correct answer when it is not. Item-level values vary around the nominal g and s on the logit scale with standard deviation 0.3. Initial mastery has logit $\text{logit}(0.35) + u_i + v_k + e_{ik}$ with $u_i \sim N(0,0.7^2)$, $v_k \sim N(0,0.5^2)$, $e_{ik} \sim N(0,0.3^2)$. After each attempt an unmastered skill becomes mastered with probability $\sigma(\text{logit}(0.18) + 0.5u_i + \omega_k)$, $\omega_k \sim N(0,0.3^2)$. Between two attempts at the same skill separated by Δ interactions, a mastered skill is forgotten with probability $1 - e^{-\lambda\Delta}$, $\lambda = 0.006$. These hand-set values give visible learning curves and moderate accuracy. In both generators accuracy rises with practice: the proportion correct is about 0.59 (KT-A) and 0.58 (KT-B) over all 60 events but 0.63 and 0.62 at the target events.

Synthetic-KT-B is a dynamic, multidimensional IRT process whose assumptions differ from BKT. Ability follows $\theta_{it} = \alpha_i + v_{i,k_t} + \beta_i\, t/(T-1) + \varepsilon_{it}$, with learner ability $\alpha_i \sim N(0,1)$, skill-specific offsets $v_{ik} \sim N(0,0.5^2)$, growth $\beta_i \sim N(1,0.4^2)$, and an AR(1) disturbance ($\phi = 0.9$, innovation SD 0.15). The response probability is

$$P(y_{it} = 1) = g_j + (1 - g_j - s_j)\,\sigma(a_j(\theta_{it} - b_j)), \tag{2}$$

with $\log(a_j) \sim N(0,0.3^2)$ and $b_j \sim N(0,1)$. Here g_j is the lower asymptote of the response curve and $1 - s_j$ its upper asymptote. The parameters g_j and s_j play the same formal role as response randomness in KT-A but are not the same psychological mechanism, so the two generators are analysed separately and never pooled. Ability drift and multidimensionality violate the assumptions of a static, unidimensional 2PL model, and item-level variation in discrimination and difficulty differs from the skill-level parameters of BKT.

Response randomness is varied through five generating configurations per mechanism: a centre configuration ($g = 0.20$, $s = 0.15$), two values of g (0.10 and 0.30, with $s = 0.15$), and two values of s (0.05 and 0.25, with $g = 0.20$). Other generator parameters were held fixed.

2.2. Prediction Task and Model Specifications

The task is next-response prediction. Given the visible history, a sequence of (item, skill, response) triples that precede the target event, and the target item, a model outputs $p = P(y = 1)$. To make every model see the same events, targets are fixed to events 50–59 of each test learner, giving ten target events per learner and 4,500 per replicate. Events are numbered from 0 to 59 throughout, so events 50–59 are a learner's last ten responses.

Five predictors are compared. (i) The *Baseline* is a smoothed item-level correct rate, $p_j = (c_j + \alpha\bar{p})/(n_j + \alpha)$, which uses no history. Here c_j and n_j are the number of correct responses and the total number of responses to item j among the training learners, $\bar{p}$ is the overall proportion correct in the training data, and α is a smoothing weight (Table 1). (ii) *BKT with forgetting* has one parameter set per skill (initial mastery, learning, forgetting, guess, slip) fitted by expectation–maximisation on training learners (up to 60 iterations from three starting points, keeping the best likelihood), with guess and slip bounded above and forgetting bounded at 0.3; the prediction filters the visible same-skill responses starting from the prior. (iii) *2PL-IRT* has item discrimination and difficulty fitted by maximum a posteriori estimation on training learners (priors: ability $N(0,1)$, log-discrimination $N(0, {\sigma_a}^2)$, difficulty $N(0,2^2)$); for a test learner the ability is re-estimated from the visible window only, with item parameters held fixed. (iv) *DKT* embeds the (skill, response) token, passes the sequence through one GRU layer (Cho et al., 2014), and outputs a probability per skill, of which the target skill's entry is used (Piech et al., 2015). (v) The *AKT-style* model uses Rasch-type question embeddings, one attention head over earlier interactions with scores multiplied by $\exp(-\vartheta d)$, where d is the number of events between the two interactions and $\vartheta > 0$ is learned, and a two-layer prediction head (Ghosh et al., 2020). It is simplified relative to the original AKT (no encoder stack, temporal rather than context-aware distance) and is therefore called AKT-style.

The predictors differ in the information they use as well as in their structure (Table 1). The baseline uses only the identity of the target item. BKT uses the target skill and the visible responses on that skill, and all items of a skill share its parameters. 2PL-IRT uses the identities of the visible and target items, through their difficulty and discrimination, and pools the visible responses of all skills into one ability. DKT uses the skill and correctness of every visible event and the target skill, but no item identities. The AKT-style model adds one Rasch-type difficulty value per item to its skill embeddings. Differences between models therefore mix structure with available information, and the comparisons below are between these implementations, not between model families in general. As a control for one source of item information, *DKT-item* keeps DKT's skill inputs but has one output per item, the combination

that worked best on most datasets in Gervet et al. (2020); its hidden size and learning rate were chosen on the same grid and development replicate as DKT's.

DKT, DKT-item, and the AKT-style model were implemented in NumPy with reverse-mode automatic differentiation, and their gradients were checked against finite differences (maximum relative error about 10^{-5}). Training used Adam (Kingma and Ba, 2015), gradient-norm clipping at 5, weight decay 10^{-5}, batch size 64, at most 40 epochs, and early stopping with patience 5 on validation loss. Hyperparameters came from small predefined grids (two to six configurations per family) evaluated on the validation learners of the development seed (Table 1; selected values for each mechanism are listed there). The simulated bank contains every item in the training data, so item cold start is not exercised. Items with several skills are not generated here; real logs would need a rule for assigning them, which changes the model inputs.

Table 1: Experimental conditions and model configurations.

Component	Setting
Learners and items	3,000 learners × 60 interactions per replicate; 200 items in 10 skills; five replicates (seeds 1–5); separate development replicate (seed 9001) for hyperparameters
Splits (by learner)	Training 60% (1,800), validation 15% (450), calibration 10% (300), test 15% (450)
Target events	Events 50–59 of each test learner (events numbered 0–59, so the last ten responses; 4,500 per replicate), identical for every condition
Generators	Synthetic-KT-A: two-state learning, forgetting, guessing, slipping (Eq. 1). Synthetic-KT-B: dynamic item response process with asymptotes (Eq. 2)
Guess g / slip s	Centre (0.20, 0.15); g = 0.10, 0.30 with s = 0.15; s = 0.05, 0.25 with g = 0.20
History length L	3, 5, 10, 20, 50 (centre configuration); 3, 10, 50 (other configurations)
History label flips η	0, 0.05, 0.10, 0.20 (centre configuration); 0, 0.10, 0.20 at L = 50 (other configurations)
Training settings	Test-only shift (all models; full-sequence loss for DKT and AKT-style); matched training with the window loss (DKT, AKT-style; centre configuration)
Control comparisons	Centre configuration: DKT-item; clean window reference (L = 50, targets 50–59); position-matched windows (L = 3–20, targets 50–59); early targets (event L with its complete history)
Baseline	Uses the target item only. Smoothed item accuracy; α = 50 (KT-A), 10 (KT-B)
BKT with forgetting	Uses the target skill and the visible responses on that skill. Per-skill initial mastery, learning, forgetting (≤ 0.3), guess, slip; upper bound on g and s 0.3 (KT-A), 0.5 (KT-B)
2PL-IRT	Uses the identities of the visible and target items and all visible responses. Item discrimination and difficulty (MAP); SD of log-discrimination 0.3 (both)
DKT	Uses the skill and response of each visible event and the target skill. GRU; token embedding 24; hidden size 32 (KT-A), 64 (KT-B); learning rate 0.01
AKT-style	Uses skills, one difficulty value per item, and all visible responses. Rasch-type embeddings 16 (both); attention with distance decay; head width 32; learning rate 0.01
DKT-item (control)	As DKT, with one output per item (the target item's entry is used). Hidden size 64 (KT-A), 16 (KT-B); learning rate 0.003
Neural training	Adam; batch 64; weight decay 1e-5; gradient clipping 5; at most 40 epochs; early stopping (patience 5) on validation NLL. Full-sequence loss: 108,000 predictions per epoch; window loss: 6,000 windows per epoch

2.3. Sparsity and Noise Conditions

Information sparsity, response randomness, and record errors are varied separately. **History length**: for each target, the visible history is the most recent $L \in \{3,5,10,20,50\}$ events in their original order. Target events are identical for every L, so a short window represents a learner with a long record of which only the last L events are visible. Each model treats the first

visible event as the start of the record: BKT starts from its initial mastery, 2PL-IRT from its ability prior, and the networks from the state they learned for the beginning of a sequence. **Label corruption**: each visible history response is flipped independently with probability $\eta \in \{0,0.05,0.10,0.20\}$ using common random numbers, so that flips are nested across η and identical across models. Only the visible history is corrupted; the target label used for scoring stays clean. **Response randomness**: guess and slip values vary at generation as described in Section 2.1 and therefore affect training, validation, and test data alike.

Two training settings are kept apart. In the *test-only shift* setting, models are fitted once on clean, full-length training histories (60 events) and evaluated under the shifted conditions; the perturbation occurs only in the test (and calibration) windows. For DKT and the AKT-style model this full-sequence training computes the loss at every position of every training learner, 108,000 supervised predictions per epoch (1,800 learners × 60 positions, the first predicted from an empty history). In the *matched training* setting, the two neural models are retrained with a window loss: each epoch draws 6,000 windows and computes the loss only at the last position of each. For history length the windows have exactly L history events and target positions anywhere from L to 59; for label corruption they have $L = 50$, their input histories are flipped, and the loss targets remain clean. Matched training therefore differs from full-sequence training in three ways at once: the inputs match the test condition, each epoch supervises about one eighteenth as many predictions, and the target positions differ. Contrasts between the two settings compare training schemes, not matching alone. Baseline, BKT, and IRT have global parameters fitted on clean training data in every condition, so these contrasts are limited to the neural models.

Three control comparisons in the centre configuration separate factors that the main design confounds (Section 3.5). For item information, DKT-item is trained and evaluated exactly like DKT. For the training scheme, a *clean window reference* is trained with the window loss on clean windows with $L = 50$ that end at events 50–59, with the same budget and stopping rule as the matched models, and is evaluated in every condition; it shares the matched models' training scheme but not their input condition. Window training matched in length is also repeated with the training targets restricted to events 50–59, the positions used at test time (*position-matched windows*). For truncated versus genuinely short histories, the test-only models are also scored on *early targets*: event L of each test learner, that is, its response number $L + 1$, with all L earlier events (events 0 to $L - 1$) visible, as for a learner who has answered only L items (450 targets per replicate).

2.4. Training, Calibration, and Evaluation Protocol

Splits are by learner, responses are ordered by time, and only earlier events enter a history; the target answer and later answers are never visible. Hyperparameter selection and early stopping use training and validation learners only, and all models share the same splits. The calibration learners (10%) are used for post-hoc calibration only; the test learners (15%) are used for reporting only. Figure 1 summarises the workflow.

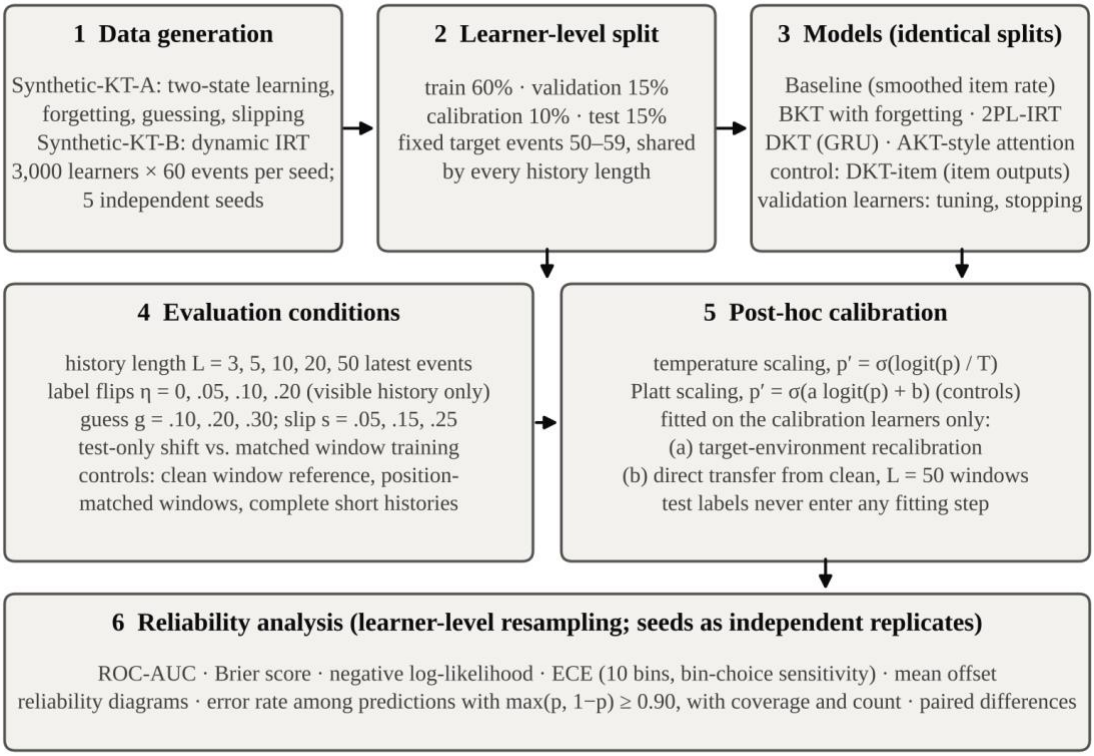


Figure 1: Study design and evaluation workflow.

Temperature scaling is used as a simple post-hoc reference. A predicted probability p is replaced by $p' = \sigma(\mathrm{logit}(p)\,/T)$ with a single $T > 0$ fitted by minimising the negative log-likelihood on the calibration learners (bounded search of $\log(T)$ over $[-2.5,2.5]$). Two uses are distinguished. In *target-environment recalibration*, T is fitted on calibration windows with the same L and η as the test windows. In *direct transfer*, T is fitted on clean windows with $L = 50$ and applied unchanged to shifted test windows. Matched-trained models are recalibrated in the target environment only. For the control comparisons, Platt scaling, $p' = \sigma(a\ \mathrm{logit}(p) + b)$ with slope a and intercept b fitted by maximum likelihood (Niculescu-Mizil and Caruana, 2005), is fitted in the same two ways, because a temperature alone has no intercept. A positive temperature is monotone, so AUC should not change; this was verified (Section 3.4).

2.5. Reliability Metrics and Statistical Analysis

Let p_i be the predicted probability of a correct response and $y_i \in \{0,1\}$ the observed response for N target events. The primary scores are the Brier score and the negative log-likelihood (NLL):

$$\mathrm{Brier} = (1/N)\sum_{i=1}^{N}(p_i - y_i)^2, \tag{3}$$

$$\mathrm{NLL} = -(1/N)\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]. \tag{4}$$

In computing NLL, probabilities are clipped to $[10^{-6}, 1-10^{-6}]$. ROC-AUC is reported as the discrimination measure. The expected calibration error (Naeini et al., 2015) uses $B = 10$ equal-width bins of the predicted probability of a correct response:

$$\mathrm{ECE} = \sum_{b=1}^{B} (n_b/N)|\bar{y}_b - \bar{p}_b|, \tag{5}$$

where n_b, $\bar{y}_b$, and $\bar{p}_b$ are the count, observed proportion correct, and mean prediction in bin b. Reliability diagrams plot $\bar{y}_b$ against $\bar{p}_b$, omitting bins with fewer than 50 pooled events. ECE is positively biased in finite samples and depends on the binning, so its sensitivity was checked with 5, 15, and 20 equal-width bins and with 10 equal-mass bins (Nixon et al., 2019), and small ECE differences are interpreted cautiously. Systematic over- or under-prediction is summarised by the *mean offset*, the mean predicted probability minus the observed proportion correct (calibration-in-the-large); it is negative when a model under-predicts.

For high-confidence errors, the predicted class is $\hat{y}_i = \mathbb{1}[p_i \geq 0.5]$ and the confidence is $c_i = \max(p_i, 1-p_i)$. Among predictions with $c_i \geq 0.90$ we report the error proportion together with the coverage (their share of all predictions) and the count; the error proportion is not available (NA) when no prediction qualifies. A smaller number of high-confidence predictions is not treated as lower error risk.

Uncertainty is quantified with learner-level resampling (Efron, 1979): the 450 test learners, with their ten target events each, are resampled with replacement, and for pooled results the per-replicate bootstrap distributions (400–500 resamples) are averaged across the five replicates before the 2.5th and 97.5th percentiles are taken. Replicate-to-replicate variability is reported as the standard deviation across the five seeds. Paired differences between models, conditions, or calibration variants use identical resampled learners, and the ten target events of one learner are never treated as independent. No significance tests or p values are reported; five replicates are too few for them to be informative.

2.6. Reproducibility

All random draws are fixed by seeds: simulation, splits, initialisation, minibatch order, label flips, and bootstrap resamples. Refitting the test-only models and the clean window reference for the control comparisons reproduced the stored test predictions of the main experiment exactly (102 prediction arrays per replicate, maximum absolute difference 0). The code, configuration files, and saved predictions, with a script that rebuilds every number, table, and figure, are available at https://github.com/jocelyn-0304/kt-predictive-reliability.

3. Predictive Reliability Analysis

3.1. Overall Predictive Performance

Table 2 reports discrimination and calibration with clean histories at the shortest ($L = 3$) and longest ($L = 50$) visible history, averaged over five replicates; intermediate lengths are in Figure 3.

Table 2: Predictive discrimination and calibration across models with clean histories ($\eta = 0$), at visible history lengths L = 3 and L = 50; mean (SD) over five replicates.

Generator	Model	L = 3				L = 50			
		AUC	Brier	NLL	ECE	AUC	Brier	NLL	ECE
KT-A	Baseline	0.543 (0.012)	0.233 (0.004)	0.658 (0.009)	0.046 (0.017)	0.543 (0.012)	0.233 (0.004)	0.658 (0.009)	0.046 (0.017)
	BKT	0.685 (0.007)	0.218 (0.004)	0.624 (0.009)	0.086 (0.015)	0.736 (0.009)	0.192 (0.002)	0.569 (0.005)	0.015 (0.001)
	2PL-IRT	0.628 (0.004)	0.225 (0.004)	0.644 (0.009)	0.062 (0.003)	0.588 (0.014)	0.231 (0.003)	0.655 (0.007)	0.061 (0.015)
	DKT	0.684 (0.007)	0.214 (0.003)	0.616 (0.006)	0.060 (0.011)	0.722 (0.010)	0.199 (0.002)	0.585 (0.006)	0.025 (0.006)
	AKT-style	0.674 (0.007)	0.221 (0.003)	0.632 (0.007)	0.074 (0.009)	0.730 (0.009)	0.196 (0.002)	0.577 (0.005)	0.024 (0.005)
KT-B	Baseline	0.629 (0.013)	0.225 (0.003)	0.642 (0.007)	0.041 (0.010)	0.629 (0.013)	0.225 (0.003)	0.642 (0.007)	0.041 (0.010)
	BKT	0.545 (0.005)	0.237 (0.003)	0.666 (0.006)	0.049 (0.009)	0.575 (0.006)	0.233 (0.003)	0.658 (0.006)	0.030 (0.009)
	2PL-IRT	0.624 (0.013)	0.232 (0.005)	0.659 (0.013)	0.068 (0.009)	0.675 (0.009)	0.217 (0.003)	0.624 (0.007)	0.044 (0.009)
	DKT	0.556 (0.009)	0.239 (0.003)	0.671 (0.006)	0.076 (0.008)	0.611 (0.005)	0.227 (0.003)	0.645 (0.006)	0.016 (0.007)
	AKT-style	0.627 (0.014)	0.228 (0.005)	0.648 (0.010)	0.052 (0.004)	0.671 (0.009)	0.215 (0.004)	0.620 (0.008)	0.018 (0.007)

AUC = ROC-AUC; NLL = negative log-likelihood; ECE = expected calibration error (10 equal-width bins). Targets are events 50–59 of 450 test learners per replicate. Lower is better for Brier, NLL, and ECE.

Take Synthetic-KT-A at $L = 50$ first. BKT, the AKT-style model, and DKT were close to one another (AUC 0.736, 0.730, 0.722; Brier 0.192, 0.196, 0.199), but not identical: paired differences computed on the same resampled learners were 0.0141 in AUC for BKT minus DKT ([0.0105, 0.0177]), 0.0060 for BKT minus the AKT-style model ([0.0024, 0.0096]), and 0.0082 for the AKT-style model minus DKT ([0.0049, 0.0117]), with paired Brier differences of −0.0068 ([−0.0080, −0.0057]), −0.0036 ([−0.0047, −0.0026]), and −0.0032 ([−0.0043, −0.0023]). These gaps are small beside the gain over the item baseline. Against the item baseline (Brier 0.233) all three gained: the paired Brier differences from the baseline were −0.041 for BKT ([−0.043, −0.038]), −0.037 for the AKT-style model ([−0.039, −0.035]), and −0.034 for DKT ([−0.036, −0.032]). 2PL-IRT sat close to the baseline (Brier 0.231; paired difference −0.0017, [−0.0036, 0.0001]), and its AUC of 0.588 was far below that of the other three.

The order changed under Synthetic-KT-B. At $L = 50$ the AKT-style model and 2PL-IRT did best (AUC 0.671 and 0.675; Brier 0.215 and 0.217), with paired Brier differences from the baseline of −0.010 ([−0.011, −0.009]) and −0.009 ([−0.011, −0.007]). BKT fared worse than the baseline on Brier score (0.233 against 0.225; paired difference +0.007, [0.005, 0.009]), and DKT showed no detectable difference from it (+0.001, [−0.001, 0.004]). The baseline's AUC of 0.629 was higher than that of BKT (0.575) and DKT (0.611). Under KT-B, items of a skill differ widely in difficulty and discrimination, and the models that represent individual items (the AKT-style model, 2PL-IRT, and the baseline) did better than the two that treat all items of a skill alike (BKT and DKT). Under KT-A, items of a skill differ only in their guess and slip values, and BKT, whose structure is closest to that generator, did best. Section 3.5 shows that DKT's gap to the AKT-style model under KT-B closes when DKT is given item outputs, so this part of the ranking reflects the information available to each implementation rather than the model family.

At $L = 50$, ECE ranged from 0.015 to 0.061 under KT-A and from 0.016 to 0.044 under KT-B, lowest for BKT (KT-A) and DKT (KT-B) and highest for 2PL-IRT in both. Figure 2 shows reliability diagrams for representative conditions; with a full history most curves lie near the diagonal, and 2PL-IRT departs from it most under KT-A.

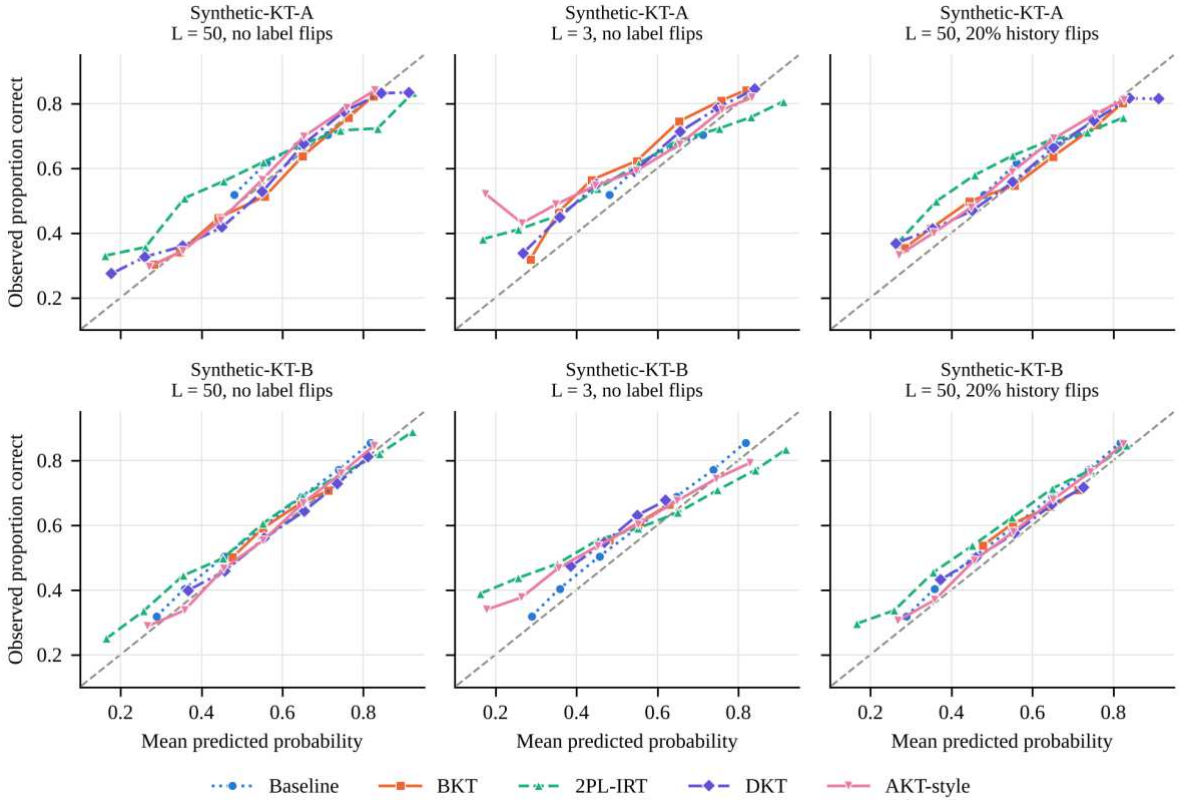


Figure 2: Reliability diagrams under representative conditions: clean full history (L = 50), three visible events (L = 3), and 20% history flips (L = 50), for Synthetic-KT-A (top) and Synthetic-KT-B (bottom). Predictions are pooled over five replicates; bins with fewer than 50 events are omitted.

3.2. Reliability Under Limited Response Histories

RQ1. Figure 3 shows that the effect of history length differed by model and generator. Under KT-A, Brier score and negative log-likelihood of BKT, DKT, and the AKT-style model improved steadily as L increased from 3 to 50 (BKT Brier 0.218 to 0.192). The paired Brier increase from $L = 50$ to $L = 3$ was 0.026 for BKT ([0.024, 0.028]), 0.015 for DKT ([0.014, 0.017]), and 0.025 for the AKT-style model ([0.023, 0.027]), and AUC fell by 0.038 to 0.056. ECE rose from 0.015–0.025 at $L = 50$ to 0.060–0.086 at $L = 3$, so at three visible events all three models had higher ECE than the baseline (0.046) although they remained better on Brier score (paired differences from the baseline −0.015 for BKT, −0.019 for DKT, −0.012 for the AKT-style model, intervals excluding zero). At $L = 3$ the observed proportion correct lay above the diagonal for predictions of roughly 0.3 to 0.7 (Figure 2, top middle): the models under-predicted correctness there, and on average their predictions were 0.060 to 0.086 below the observed proportion correct (Section 3.5).

The least intuitive result in this section concerns 2PL-IRT under KT-A: its AUC was 0.628 at $L = 3$ and 0.588 at $L = 50$, and its Brier score was 0.225 and 0.231 (paired difference $L = 3$ minus $L = 50$: −0.006, [−0.008, −0.003]). A longer window was worse for this model. That fits a static ability estimate being applied to a learner whose mastery changes inside the window, which the KT-A generator produces. Its mean offset was also larger at $L = 50$ (−0.041) than at $L = 3$ (−0.031).

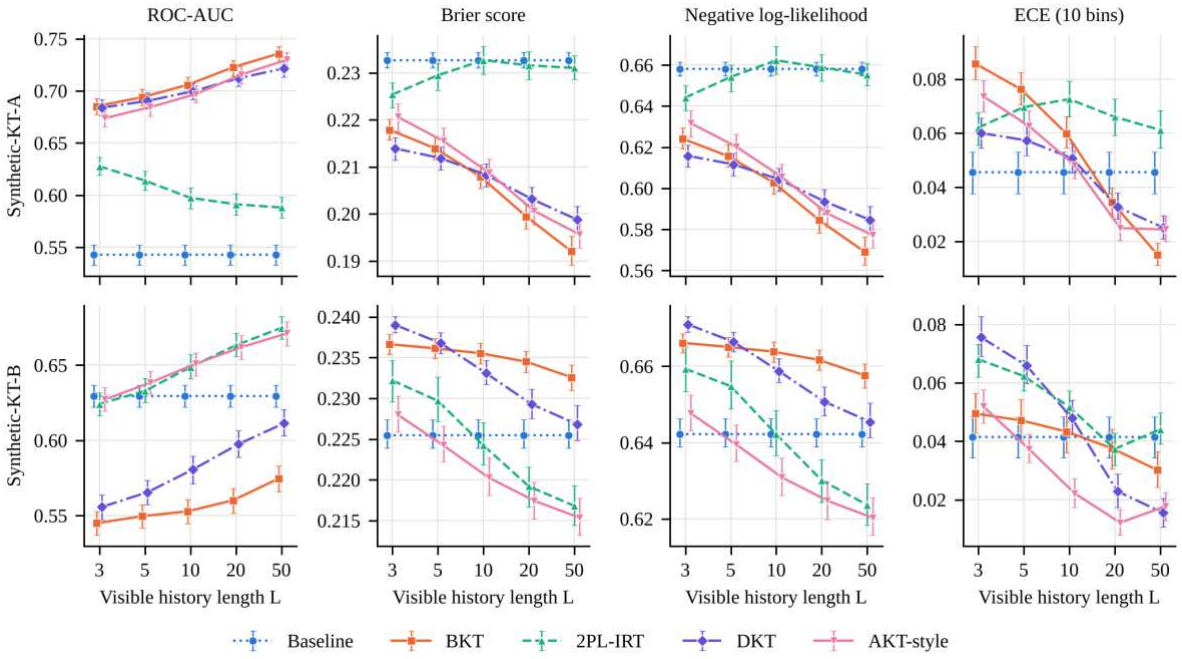


Figure 3: Change in AUC, Brier score, NLL, and ECE with visible history length (η = 0). Points are means of the five replicate estimates; bars are 95% learner-level bootstrap half-widths.

Under KT-B every non-baseline model improved with longer histories, but at $L = 3$ none beat the item baseline on Brier score: the paired differences were +0.003 for the AKT-style model ([0.001, 0.004]), +0.007 for 2PL-IRT ([0.005, 0.009]), +0.011 for BKT, and +0.014 for DKT. The point estimates of Brier score crossed the baseline at $L = 5$ for the AKT-style model and at $L = 10$ for 2PL-IRT. Baseline ECE (0.041) was also lower than that of every other model at $L = 3$ (0.049 to 0.076). ECE at $L = 50$ was 0.016 to 0.044, and the AKT-style model reached its lowest value (0.012) at $L = 20$, so ECE need not decrease monotonically with L over this grid.

Matched training, available for the two neural models only (Section 2.3), changed these patterns. At $L = 3$ and $L = 5$, matched models had lower ECE than their test-only counterparts in all eight model–generator–length combinations; at $L = 3$ the paired ECE differences were −0.019 ([−0.023, −0.015]) for DKT and −0.035 ([−0.040, −0.029]) for the AKT-style model under KT-A, and −0.039 ([−0.040, −0.037]) and −0.020 ([−0.024, −0.015]) under KT-B. Brier differences at $L = 3$ were small (−0.0075 to −0.0005). At $L = 50$ the matched model sees the same input as the test-only model and differs from it only in the training scheme; it was worse on Brier score (+0.004 to +0.013) and AUC (−0.015 to −0.040). This is a cost of the window loss, which supervises about one eighteenth as many predictions per epoch, rather than of matching; Section 3.5 separates the two.

3.3. Reliability Under Guessing, Slipping, and Label Corruption

RQ2. Varying response randomness at generation changed discrimination more than calibration. At $L = 50$ under KT-A, BKT AUC was 0.775, 0.736, and 0.698 for $g = 0.10$, 0.20, and 0.30, and 0.818, 0.736, and 0.677 for $s = 0.05$, 0.15, and 0.25; Brier scores moved in the opposite direction (0.148 at $s = 0.05$ to 0.221 at $s = 0.25$). At $s = 0.25$ the margin over the baseline narrowed under both generators: under KT-B the AKT-style model (Brier 0.233) and 2PL-IRT (0.234) were only slightly below the baseline (0.239), while BKT (0.246) and DKT (0.242) were above it. ECE at $L = 50$ showed no clear trend with g or s (for example 0.013–0.015 for BKT and 0.020–0.025 for DKT under KT-A).

High-confidence errors are summarised in Table 3 (panel A) as the pooled error proportion among predictions with confidence of at least 0.90, with the number of such predictions out of 22,500 per condition. With default g and s and a clean full history they were rare or absent: none for the baseline and BKT under either generator, and 82 (0.4%), 392 (1.7%), and 13 for 2PL-IRT, DKT, and the AKT-style model under KT-A. Error proportions for 2PL-IRT (0.159 under KT-A, 0.120 under KT-B) and DKT (0.166 under KT-A) were above 0.10, the maximum expected of a calibrated predictor at this confidence level, but the counts are small (for example 13 errors among 82) and no interval was computed. With three visible events, 2PL-IRT produced 77 such predictions under KT-A (error 0.182) and 342 under KT-B (0.187), and the AKT-style model 33 under KT-A (0.303; 10 errors). These figures flag where overconfidence may occur; they do not estimate its size.

Table 3: High-confidence errors and calibration changes under sparse or noisy conditions. Panel A: high-confidence predictions (confidence ≥ 0.90).

Model	**L = 50, η = 0**		**L = 3, η = 0**		**L = 50, η = 0.20**		**L = 50, η = 0, s = 0.05**	
	n	**Error**	**n**	**Error**	**n**	**Error**	**n**	**Error**
Synthetic-KT-A								
Baseline	0	NA	0	NA	0	NA	0	NA
BKT	0	NA	0	NA	0	NA	9,915	0.079
2PL-IRT	82	0.159	77	0.182	3	0.000	590	0.159
DKT	392	0.166	0	NA	177	0.186	6,896	0.065
AKT-style	13	0.154	33	0.303	8	0.125	5,451	0.070
Synthetic-KT-B								
Baseline	0	NA	0	NA	0	NA	65	0.062
BKT	0	NA	0	NA	0	NA	0	NA
2PL-IRT	266	0.120	342	0.187	22	0.136	1,701	0.076
DKT	0	NA	0	NA	0	NA	34	0.118
AKT-style	2	0.000	18	0.056	2	0.000	803	0.052

Panel A. n = number of predictions with confidence max(p, 1 − p) ≥ 0.90 among 22,500 target events (five replicates pooled); Error = proportion of these predictions whose predicted class was wrong; NA = no such prediction. Coverage is n / 22,500. Baseline is the smoothed item-accuracy model.

Panel B. Expected calibration error (10 bins) before and after temperature scaling.

Model	L = 3, η = 0			L = 3, η = 0.20			L = 50, η = 0.20		
	Raw	TS-target	TS-ref	Raw	TS-target	TS-ref	Raw	TS-target	TS-ref
Synthetic-KT-A									
Baseline	0.046	0.029	0.029	0.046	0.029	0.029	0.046	0.029	0.029
BKT	0.086	0.086	0.087	0.100	0.104	0.101	0.033	0.024	0.031
2PL-IRT	0.062	0.054	0.054	0.083	0.081	0.078	0.080	0.080	0.085
DKT	0.060	0.062	0.062	0.077	0.081	0.077	0.030	0.028	0.028
AKT-style	0.074	0.076	0.081	0.103	0.093	0.116	0.030	0.029	0.034
Synthetic-KT-B									
Baseline	0.041	0.033	0.033	0.041	0.033	0.033	0.041	0.033	0.033
BKT	0.049	0.025	0.033	0.054	0.035	0.041	0.039	0.034	0.034
2PL-IRT	0.068	0.054	0.064	0.087	0.076	0.085	0.068	0.067	0.069
DKT	0.076	0.049	0.075	0.082	0.064	0.081	0.020	0.020	0.020
AKT-style	0.052	0.052	0.055	0.070	0.070	0.072	0.027	0.020	0.019

Panel B. ECE, mean over five replicates. Raw = unscaled; TS-target = temperature fitted on calibration windows with the same L and η as the test windows; TS-ref = temperature fitted on clean L = 50 windows and applied unchanged. The baseline ignores history, so its TS-target and TS-ref columns coincide.

At $s = 0.05$ high-confidence predictions became common, with error proportions of 0.079 for BKT (9,915 predictions, 44% of events), 0.065 for DKT (6,896), and 0.070 for the AKT-style model (5,451) under KT-A, and 0.076 for 2PL-IRT (1,701) and 0.052 for the AKT-style model (803) under KT-B; 2PL-IRT under KT-A stayed at 0.159 (590). At $s = 0.25$ and $g = 0.30$ the high-confidence set was empty or nearly so for almost every model, so its size depends strongly on the response-randomness setting.

History label flips reduced discrimination and worsened Brier score at both history lengths. With $L = 50$ and test-only flips, under KT-A the AUC of BKT, DKT, and the AKT-style model fell from 0.736, 0.722, and 0.730 at $\eta = 0$ to 0.685, 0.679, and 0.683 at $\eta = 0.20$, and Brier scores rose from 0.192, 0.199, and 0.196 to 0.210, 0.212, and 0.210; 2PL-IRT moved from 0.231 to 0.235. Under KT-B the changes were smaller (2PL-IRT Brier 0.217 to 0.223; AKT-style 0.215 to 0.219; both still below the baseline's 0.225). ECE increased with η for BKT under KT-A (0.015 to 0.033) and for 2PL-IRT under both generators (0.061 to 0.080; 0.044 to 0.068). At $L = 3$ with $\eta = 0.20$, sparsity and noise together removed the advantage over the baseline in most cases: under KT-A the Brier scores of BKT (0.234), 2PL-IRT (0.236), and the AKT-style model (0.238) were not below the baseline (0.233), and only DKT (0.229) was; under KT-B every model was above it (0.234 to 0.242 against 0.225).

Flips mainly reduced the number of high-confidence predictions, and the error proportion among them did not change consistently (Table 3, panel A). For DKT under KT-A at $L = 50$, the pooled error proportion rose from 0.166 to 0.186 as coverage fell from 1.7% to 0.8% of events (392 to 177 predictions), and for 2PL-IRT under KT-B it rose from 0.120 to 0.136 (266 to 22 predictions); for 2PL-IRT under KT-A only 3 predictions remained at $\eta = 0.20$. The mean of the five replicate-level proportions for DKT was 0.19 to 0.21 across η, with a standard deviation of 0.06 to 0.07 between replicates, so with counts this small the direction of the change is not established. Matched training on flipped input histories with clean targets gave no consistent gains at $\eta = 0.20$: Brier scores at $L = 50$ were 0.219 (DKT) and 0.220 (AKT-style) under KT-A, against 0.212 and 0.210 for the test-only versions, and 0.232 and 0.222 under KT-B, against 0.230 and 0.219. Most of this gap reflects the window loss itself: against the clean window reference, which shares the matched models' training scheme, matching the noise helped DKT but not the AKT-style model (Section 3.5).

3.4. Calibration, High-Confidence Errors, and Sensitivity Analyses

RQ3. Temperature scaling did not change AUC, as expected for a monotone transformation. With a clean full history, fitted temperatures were between 0.79 and 1.19 for every model except the baseline (0.72 under KT-A, 0.83 under KT-B), and the paired changes after scaling were small. The baseline's ECE fell from 0.046 to 0.029 under KT-A (paired change −0.016, [−0.022, −0.010]) and the AKT-style model's by 0.006 ([−0.010, −0.001]); for BKT and DKT under KT-A the intervals for the changes in negative log-likelihood and ECE included zero.

At $L = 3$ the picture was more varied (Table 3, panel B). Refitting the temperature on calibration windows of the same length (target-environment recalibration) reduced negative log-likelihood under KT-B for 2PL-IRT (−0.011, [−0.012, −0.009]), DKT (−0.004, [−0.006, −0.003]), and BKT (−0.003, [−0.004, −0.001]), and reduced ECE for the same models (0.068 to 0.054, 0.076 to 0.049, 0.049 to 0.025). Under KT-A, where raw ECE at $L = 3$ was 0.060 to 0.086 for BKT, DKT, and the AKT-style model, recalibration lowered negative log-likelihood only for 2PL-IRT and the AKT-style model (−0.004 and −0.005) and did not detectably change ECE for BKT, DKT, or the AKT-style model (intervals include zero). The under-prediction visible in Figure 2 is mainly an offset, which a single temperature, having no intercept, could not remove; Section 3.5 shows that a map with an intercept could.

Transfer of the temperature from the clean $L = 50$ reference was weaker and sometimes harmful. The fitted values differed strongly between environments (DKT under KT-B: 0.98 on clean long histories against 0.54 at $L = 3$; AKT-style under KT-A: 0.91 against 1.24; 2PL-IRT under KT-B: 1.04 against 1.51). Direct transfer lowered negative log-likelihood for DKT under KT-B by only 0.0002 (target-environment: 0.0043) and for 2PL-IRT by 0.002 (0.011), and it raised negative log-likelihood and ECE for the AKT-style model at $L = 3$ under KT-A (+0.0055, [0.0048, 0.0061]; ECE +0.0077, [0.0044, 0.0110]) and under KT-B (+0.0023 and +0.0032). With $L = 50$ and $\eta = 0.20$ the gains remained small; for BKT under KT-A, target-environment recalibration lowered ECE by 0.009 ([−0.015, −0.003]) and direct transfer by 0.001 ([−0.003, 0.001]). Direct transfer lowered ECE in some cases (BKT under KT-B at $L = 3$ and $\eta = 0$: 0.049 to 0.033) and raised it in others, so the benefit of this simple correction did not transfer consistently across history lengths and noise levels, and the largest benefits were small in absolute terms (negative log-likelihood below 0.011, ECE below 0.027).

After scaling, high-confidence error proportions were defined for fewer settings and rested on very few events (many cells were NA), so we draw no conclusion on whether temperature scaling reduces high-confidence errors. Rankings by ECE were checked with 5, 10, 15, and 20 equal-width bins and 10 equal-mass bins in six representative conditions (both generators at $L = 50$, at $L = 3$, and at $L = 50$ with $\eta = 0.20$). The best and worst models were stable, but in five of the six conditions at least two adjacent models swapped places under some binning; for example, DKT and the AKT-style model under KT-A at $L = 50$ swapped with 20 bins (0.027 against 0.029). ECE differences of a few thousandths should therefore not be interpreted. Replicate variability (SD across the five replicates) at $L = 50$ was 0.005–0.014 for AUC, 0.002–0.004 for Brier score, and up to 0.017 for ECE (Table 2).

3.5. Control Comparisons

Item information. Giving DKT one output per item changed its standing under KT-B but not under KT-A (Table 4, panel A). Under KT-B, DKT-item raised AUC over DKT by 0.0595 at $L = 50$ ([0.0518, 0.0670]) and by 0.081 at $L = 3$ ([0.072, 0.090]) and lowered Brier score by about 0.011 at both lengths. It was then level with the AKT-style model (paired Brier

difference +0.0001, [−0.0005, 0.0007], at $L = 50$) and, like it, better than the baseline at $L = 50$ (−0.010, [−0.011, −0.009]) but not at $L = 3$ (+0.0026, [0.0016, 0.0035]). Under KT-A, where items of a skill differ only in their guess and slip values, item outputs lowered AUC at $L = 50$ (−0.023, [−0.027, −0.019]), raised Brier score (+0.008, [0.007, 0.009]), and made no detectable difference to either at $L = 3$. DKT's weak results under KT-B in Sections 3.1 and 3.2 therefore reflect its skill-level outputs more than its recurrent structure.

Truncated and complete short histories. With three visible events from a long record, every history-based model under-predicted on average (Table 4, panel B): the mean offset ranged from −0.031 (2PL-IRT) to −0.086 (BKT) under KT-A and from −0.026 to −0.076 under KT-B, and it was smaller at $L = 50$ than at $L = 3$ for every history-based model except 2PL-IRT (for DKT under KT-A, −0.060, −0.045, and −0.001 at $L = 3$, 10, and 50). When the same models predicted event 3 of each test learner (its fourth response, after events 0–2), so that the three visible events were the learner's complete record, the mean offsets lay between −0.011 and +0.026. The few visible events were therefore not the problem in themselves. A truncated window starts each model from a state that describes earlier practice: BKT's initial mastery and the networks' starting state describe the beginning of a sequence, and 2PL-IRT's ability prior describes a learner's average over the whole sequence. A learner at events 50–59 has practised for 50 events and answers correctly more often (proportion correct 0.635 against 0.522 at event 3 under KT-A). ECE is not compared between the two cases, because early targets give one event per learner and ECE is biased upward in small samples.

Table 4: Control comparisons in the centre configuration. Panel A: item information (test-only models, clean histories; mean (SD) over five replicates).

Model	**L = 3**			**L = 50**		
	AUC	**Brier**	**ECE**	**AUC**	**Brier**	**ECE**
Synthetic-KT-A						
DKT	0.684 (0.007)	0.214 (0.003)	0.060 (0.011)	0.722 (0.010)	0.199 (0.002)	0.025 (0.006)
DKT-item	0.681 (0.005)	0.214 (0.003)	0.054 (0.010)	0.698 (0.008)	0.207 (0.004)	0.024 (0.006)
AKT-style	0.674 (0.007)	0.221 (0.003)	0.074 (0.009)	0.730 (0.009)	0.196 (0.002)	0.024 (0.005)
Synthetic-KT-B						
DKT	0.556 (0.009)	0.239 (0.003)	0.076 (0.008)	0.611 (0.005)	0.227 (0.003)	0.016 (0.007)
DKT-item	0.637 (0.015)	0.228 (0.004)	0.074 (0.011)	0.671 (0.012)	0.215 (0.004)	0.014 (0.002)
AKT-style	0.627 (0.014)	0.228 (0.005)	0.052 (0.004)	0.671 (0.009)	0.215 (0.004)	0.018 (0.007)

Panel A. DKT-item = DKT with skill inputs and one output per item. Targets are events 50–59 with the last L events visible. AUC = ROC-AUC; ECE = expected calibration error (10 bins).

Panel B. Three visible events (L = 3, η = 0): mean offset and ECE (means over five replicates).

Model	Mean offset				ECE			
	Truncated	Complete	Matched	Position-matched	Raw	TS-target	Platt-target	Platt-ref
Synthetic-KT-A								
Baseline	−0.045	0.068	–	–	0.046	0.029	0.011	0.011
BKT	−0.086	−0.002	–	–	0.086	0.086	0.018	0.092
2PL-IRT	−0.031	0.011	–	–	0.062	0.054	0.017	0.019
DKT	−0.060	0.018	−0.035	−0.001	0.060	0.062	0.020	0.059
AKT-style	−0.067	0.010	−0.034	0.007	0.074	0.076	0.022	0.073
DKT-item	−0.050	0.026	–	–	0.054	0.054	0.019	0.040
Synthetic-KT-B								
Baseline	−0.041	0.035	–	–	0.041	0.033	0.016	0.016
BKT	−0.049	0.023	–	–	0.049	0.025	0.013	0.014
2PL-IRT	−0.026	0.007	–	–	0.068	0.054	0.015	0.050
DKT	−0.076	−0.007	−0.036	−0.010	0.076	0.049	0.014	0.066
AKT-style	−0.046	0.004	−0.028	0.012	0.052	0.052	0.014	0.043
DKT-item	−0.074	−0.011	–	–	0.074	0.063	0.015	0.058

Panel B. Mean offset = mean predicted probability minus observed proportion correct (negative = under-prediction). Truncated: targets are events 50–59 and only the three events before each target are visible (test-only models). Complete: the target is event 3 of each test learner (its fourth response; events are numbered from 0), and events 0–2 are its complete record. Matched and Position-matched: window loss with three history events and training targets at events 3–59 or at events 50–59 (neural models only; a dash marks cells that do not apply). ECE columns refer to the test-only models on truncated windows: Raw = unscaled; TS = temperature scaling; Platt = Platt scaling; target = fitted on calibration windows of the same condition; ref = fitted on clean L = 50 windows.

Training scheme and matching. The window-trained variants share one loss and budget (6,000 windows per epoch). At $L = 3$, matching the window length reduced the offset of the test-only models (DKT under KT-A: −0.060 to −0.035), and restricting the training targets to the test positions removed it (−0.001); ECE fell from 0.060 to 0.041 and then to 0.023. The same order held for the AKT-style model and under KT-B, where the position-matched offsets lay between −0.010 and +0.012 (Table 4, panel B). Brier score hardly changed (position-matched minus matched windows: −0.0001 to +0.0017) and AUC was slightly lower (by 0.005 to 0.017), so the gain is in calibration rather than in ranking. For label noise, the matched models were compared with the clean window reference, which shares their training scheme. At $L = 50$ and $\eta = 0.20$, matching lowered DKT's Brier score under both generators (KT-A −0.006, [−0.007, −0.004]; KT-B −0.002, [−0.003, −0.001]) and its ECE (−0.019 and −0.016), but it raised the AKT-style model's Brier score under KT-A (+0.004, [0.002, 0.005]) and made no detectable difference under KT-B. The clean window reference was itself worse than the full-sequence models at $L = 50$ (Brier +0.004 to +0.013), so the comparisons of matched and test-only models in Sections 3.2 and 3.3 mainly measure the cost of the window loss.

Recalibration with an intercept. Platt scaling fitted in the target condition removed the offset that a temperature mostly left in place. At $L = 3$, temperature scaling changed the mean offsets of the history-based models little or enlarged them, except for BKT under KT-B (−0.049 to −0.010) and DKT under KT-B (−0.076 to −0.040), while Platt scaling brought every offset within 0.011 of zero. ECE after Platt scaling was 0.017–0.022 under KT-A and 0.013–0.015 under KT-B, against 0.054–0.086 and 0.025–0.063 after temperature scaling; the paired ECE differences between the two maps ranged from −0.032 to −0.064 under KT-A and from −0.012 to −0.044 under KT-B, all with intervals excluding zero. The fitted maps show why: for BKT, DKT, and the AKT-style model under KT-A the intercept was 0.31 to 0.41 on the logit scale at $L = 3$ but between −0.03 and 0.05 at $L = 50$. Carried over from clean $L =$

50 windows, a Platt map therefore helped most where the model was already offset at $L = 50$ (2PL-IRT under KT-A, ECE 0.062 to 0.019; BKT under KT-B, 0.049 to 0.014) and changed ECE little or raised it for BKT, DKT, and the AKT-style model under KT-A (0.059 to 0.092 after transfer).

4. DISCUSSION

The comparisons do not single out a best model, and the control comparisons show why: model rankings were sensitive to the generating assumptions and to the information each implementation used, and most of the calibration problems traced back to a few mechanisms rather than to the model families themselves.

Item information decided part of the ranking. Under KT-B, items of a skill differ widely in difficulty and discrimination, and the models that represent individual items did well while BKT and DKT, which treat the items of a skill alike, did not. Giving DKT item outputs closed its gap to the AKT-style model, in line with the observation of Gervet et al. (2020) that DKT's results depend on whether it works with skills or items. Under KT-A, where items of a skill differ only in their guess and slip values, the same change cost discrimination; a plausible reason is that two hundred item outputs must each be learned from far fewer responses than ten skill outputs. A comparison of model families on real logs therefore needs to state what each implementation knows about items, and a skill-level model should be compared with an item-aware counterpart when items of a skill differ.

The largest calibration effect came from how a model starts a truncated window. Each history-based model treats the first visible event as the start of a learner's record and begins from a state that describes earlier practice. When the window holds the last few events of a long record, that state lags behind the learner and the model under-predicts. For every model except 2PL-IRT the offset shrank as the window grew; it was largely absent when the same few events were a learner's complete record, and it disappeared when the window loss drew its training targets from the test positions. The short-history results of RQ1 are therefore results about truncated records; genuinely new learners pose a different problem, without this offset. The mechanism also suggests remedies, not all tested here: carry a learner's state or practice count across windows, train on windows drawn from the same stage of practice as the predictions, or recalibrate with an intercept in the condition where the model is used.

The same mechanism explains the RQ3 results. A temperature rescales logits around zero, so it moves the mean prediction only as a side effect of sharpening or flattening; it left the truncation offset in place for most models, and a temperature fitted on long histories, where the offset is small, could not anticipate it. Platt scaling has an intercept and removed the offset when fitted in the target condition, but a Platt map carried over from long histories helped mainly where the model was already offset with long histories. A calibration map describes one miscalibration and transfers only to conditions that share it.

The training-scheme controls change how the matched-training results should be read. The window loss supervises about one eighteenth as many predictions per epoch as full-sequence training, and it cost discrimination even when the input matched the test condition exactly. Held to the same scheme, matching helped: windows of the test length whose targets came from the test positions lowered ECE for both networks under both generators, and flipped training histories helped DKT under both generators, though not the AKT-style model. The weaker results of matched training against full-sequence training therefore do not show that matching is ineffective; they show that the window loss is a costly way to obtain it. A loss that

keeps every position of a sequence while truncating or corrupting the history visible at each position would avoid this trade-off.

Two properties of the generators shaped the remaining differences. Both let accuracy rise with practice, which is what makes truncation costly; the offset should be smaller where learners change little, although that was not tested here. Within KT-A, mastery also changes inside a 50-event window, so a static ability estimated from the whole window lags behind the learner's current state, which is why 2PL-IRT discriminated better with three events than with fifty and was more offset with fifty. Label flips add apparent guessing and slipping to the visible history; they reduce what a window reveals, and with short windows they left the history-based models no better than the item baseline in most cases.

For evaluation practice the results suggest four habits: report calibration, including the mean offset, next to AUC; describe what each model knows about items; state whether the visible history is a learner's complete record or a truncated window; and say which recalibration map was used and whether it was fitted in the deployment condition or carried over.

Several methodological limitations bound the scope of these findings. All data were simulated from two generators with specified parametric forms, so the observed orderings reflect these processes and require empirical validation on real learner logs before any use with learners. Two generators cannot show how widely the mechanisms hold; the truncation offset depends on how much accuracy changes with practice, and both generators let it rise. The generators contain no multi-skill items, no cold-start items, no timing or help requests, and no non-random missingness. The AKT-style model is simplified, the networks are small, and only one item-aware variant of DKT was tried. Hyperparameters were chosen once, on a separate development replicate with clean full-length histories. The study uses five replicates, and inference rests on learner-level bootstrap intervals rather than formal significance tests. Flipped labels are symmetric and random, whereas real record errors are probably systematic. The high-confidence analysis rests on small counts, and ECE is biased and depends on binning. Guessing and slipping were varied only at generation, so matched and test-only effects could not be separated for them. The control comparisons, including Platt scaling, cover the centre configuration only.

The obvious next step is to repeat the protocol on public logs such as ASSISTments (Feng et al., 2009) and EdNet (Choi et al., 2020), where the position of each response within a learner's logged record is known, so truncated and complete short histories can be told apart, and to compare training losses that keep every position while truncating the visible history, as well as models with explicit uncertainty. Work with particular learner groups or classrooms would need separate designs and, where learners are involved, ethical review.

5. CONCLUSION

We asked how short and noisy histories affect the probabilities that five next-response predictors assign, using simulated learners from two generators and control comparisons that separate model family from item information, training scheme from matching, and truncated from complete short histories. For RQ1, shorter visible histories lowered AUC and raised Brier score and ECE for most models, and model rankings depended on the generator and on whether a model used item information; the calibration loss came mainly from truncation, which made every history-based model under-predict, rather than from the small number of visible events. For RQ2, flipped history labels reduced discrimination and, together with short

histories, removed the advantage over the baseline in several settings. High-confidence predictions were rare except at low slip; where they did occur, the error proportions were in several cases above the maximum error rate of 0.10 expected of a calibrated predictor at this threshold, though on small counts. For RQ3, temperature scaling changed little when histories were long, helped mainly under one generator at short histories, and left the truncation offset in place for most models; Platt scaling fitted in the target condition removed it, and maps carried over from long histories helped mainly where the model was already offset with long histories.

These conclusions are limited to the simulated settings, these implementations, and these metrics. The contribution is a controlled comparison that exposes the truncation mechanism, together with a reproducible protocol, code, and saved predictions for reporting discrimination, calibration, and high-confidence errors under sparse and noisy histories.

Declaration of Generative AI Software Tools in the Writing Process

During the preparation of this work, the author used Claude (Anthropic) as an assistive tool for English language editing, manuscript formatting, and code debugging and verification. The experimental design, mathematical formulations, statistical analyses, and scientific conclusions are the original work of the author. The author reviewed and verified all text and code and takes full responsibility for the content of the publication.

References

Abdelrahman, G., Wang, Q., and Nunes, B. P. 2022. *Knowledge tracing: A survey*. arXiv. https://doi.org/10.48550/arXiv.2201.06953

Baker, R. S. J. d., Corbett, A. T., and Aleven, V. 2008. More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian knowledge tracing. In *Intelligent Tutoring Systems, ITS 2008* (Lecture Notes in Computer Science, Vol. 5091, pp. 406–415). Springer. https://doi.org/10.1007/978-3-540-69132-7_44

Beck, J. E., and Chang, K. 2007. Identifiability: A fundamental problem of student modeling. In *User Modeling 2007* (Lecture Notes in Computer Science, Vol. 4511, pp. 137–146). Springer. https://doi.org/10.1007/978-3-540-73078-1_17

Cheng, W., Du, H., Li, C., Ni, E., Tan, L., Xu, T., and Ni, Y. 2025. Uncertainty-aware knowledge tracing. *Proceedings of the AAAI Conference on Artificial Intelligence 39*, 27, 27905–27913. https://doi.org/10.1609/aaai.v39i27.35007

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1724–1734). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1179

Choi, Y., Lee, Y., Shin, D., Cho, J., Park, S., Lee, S., Baek, J., Bae, C., Kim, B., and Heo, J. 2020. EdNet: A large-scale hierarchical dataset in education. In *Artificial Intelligence in Education, AIED 2020* (Lecture Notes in Computer Science, Vol. 12164, pp. 69–73). Springer. (Preprint: arXiv:1912.03072)

Christie, S. T., Cook, C., and Rafferty, A. N. 2024. Uncertainty-preserving deep knowledge tracing with state-space models. In *Proceedings of the 17th International Conference on Educational Data Mining* (pp. 909–914). International Educational Data Mining Society. https://doi.org/10.5281/zenodo.12729995

Corbett, A. T., and Anderson, J. R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction 4*, 4, 253–278. https://doi.org/10.1007/BF01099821

Efron, B. 1979. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics 7*, 1, 1–26. https://doi.org/10.1214/aos/1176344552

Feng, M., Heffernan, N., and Koedinger, K. 2009. Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction 19*, 3, 243–266. https://doi.org/10.1007/s11257-009-9063-7

Frénay, B., and Verleysen, M. 2014. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems 25*, 5, 845–869. https://doi.org/10.1109/TNNLS.2013.2292894

Gervet, T., Koedinger, K., Schneider, J., and Mitchell, T. 2020. When is deep learning the best approach to knowledge tracing? *Journal of Educational Data Mining 12*, 3, 31–54. https://doi.org/10.5281/zenodo.4143614

Ghosh, A., Heffernan, N., and Lan, A. S. 2020. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2330–2339). ACM. https://doi.org/10.1145/3394486.3403282

Gu, Z., Chang, R., Li, C., He, Q., Zhao, Z., and Tao, M. 2026. Uncertainty-aware hierarchical graph transformers for adaptive resource optimization in service meshes. In *2026 International Conference on Communication Networks and Machine Learning (CNML)* (pp. 1351–1354). https://doi.org/10.1109/cnml68938.2026.11452354

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning* (PMLR Vol. 70, pp. 1321–1330).

Khajah, M., Lindsey, R. V., and Mozer, M. C. 2016. How deep is knowledge tracing? In *Proceedings of the 9th International Conference on Educational Data Mining*. International Educational Data Mining Society.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*. arXiv:1412.6980

Liu, Z., Liu, Q., Chen, J., Huang, S., Tang, J., and Luo, W. 2022. pyKT: A Python library to benchmark deep learning based knowledge tracing models. In *Advances in Neural Information Processing Systems* (Datasets and Benchmarks Track, Vol. 35).

Mitton, J., Bhattacharyya, P., Abboud, R., and Woodhead, S. 2026. Knowing when to defer: Selective prediction for responsible knowledge tracing. In *Proceedings of the Impactful and Responsible AI Systems for Education Workshop* (PMLR Vol. 339, pp. 22–42).

Naeini, M. P., Cooper, G. F., and Hauskrecht, M. 2015. Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. AAAI Press.

Niculescu-Mizil, A., and Caruana, R. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning* (pp. 625–632). ACM. https://doi.org/10.1145/1102351.1102430

Nixon, J., Dusenberry, M. W., Zhang, L., Jerfel, G., and Tran, D. 2019. Measuring calibration in deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 38–41).

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J. V., Lakshminarayanan, B., and Snoek, J. 2019. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems* (Vol. 32).

Pelánek, R. 2015. Metrics for evaluation of student models. *Journal of Educational Data Mining 7*, 2, 1–19. https://doi.org/10.5281/zenodo.3554665

Pelánek, R. 2017. Bayesian knowledge tracing, logistic models, and beyond: An overview of learner modeling techniques. *User Modeling and User-Adapted Interaction 27*, 3–5, 313–350. https://doi.org/10.1007/s11257-017-9193-2

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., and Sohl-Dickstein, J. 2015. Deep knowledge tracing. In *Advances in Neural Information Processing Systems* (Vol. 28, pp. 505–513).

Tao, M. 2026. Anomaly-aware resource scheduling in LLM inference & serving. In *Proceedings of the 2026 5th International Conference on Cyber Security, Artificial Intelligence and Digital Economy* (pp. 376–382). https://doi.org/10.1145/3819836.3819894