PREDICTION IS NOT SIMULATION: PROCESS CONTRACTS FOR AUDITING AND REPAIRING KNOWLEDGE-TRACING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge-tracing models are selected as next-response predictors but reused as simulators for instructional decisions. Predictive fidelity on logged data does not identify how a model changes state under a controlled curriculum. This paper introduces *process contracts* as a second validation axis. They specify directional or practically null responses to matched trajectory transformations. The matched audit swaps early concept blocks while fixing the learner prefix, item–response multiset, exposure dose, delayed suffix, and probes. A contract passes only when its whole 95% interval lies on the prescribed side of zero or inside a predeclared ± 0.01 band. On ASSISTments 2009 and a four-regime simulator, process contracts expose transition semantics that vary with architecture, seed, and logging policy and that test AUC does not determine. In a fresh 600-checkpoint matrix, the oracle and DKT pass all 24 endpoint contracts, whereas SAKT passes 16. Four of its eight failures occur at a mean AUC of at least 0.85. The paper also introduces Process-Contrastive Regularization (PCR), a hinge penalty added to the factual cross-entropy. The penalty pushes matched learner-level effects past a margin or into the band and uses item identities that the audit never scores. The promotion gate requires a matched test-AUC loss of at most 0.01 and strictly fewer failed contracts than the matched baseline. PCR passes this gate in an exploratory single-seed pilot (from 4/16 to 1/16) and for SAKT across fresh seeds under three logging policies. The gate refuses PCR trained against a misspecified contract.

1 INTRODUCTION

Prediction and simulation impose different correctness requirements. A next-response predictor estimates what happens under the logging distribution, whereas an instructional planner asks what happens after a curriculum chosen by a new policy. Logged loss constrains the first question but can leave the second underdetermined. Models can therefore agree on observed histories yet encode different state transitions after the same controlled intervention. This gap matters because knowledge-tracing architectures represent history through recurrent memory or attention with different inductive biases (Piech et al., 2015; Zhang et al., 2017; Pandey & Karypis, 2019; Ghosh et al., 2020; Liu et al., 2023a). Reported gaps between those families also shrink once the evaluation protocol is fixed (Kim & Kim, 2026), so a predictive ranking alone is not a stable basis for promotion.

The missing requirement is *process validity*, the property that a learned model satisfies a prespecified process contract under a controlled instructional process. A process contract is neither another predictive metric nor a claim about human causality. It is an executable statement of simulator semantics. For example, a contract can require that swapping two early concept blocks produce a positive target response, a negative source response, or an effect inside a practical null band. This formulation turns implicit architectural behavior into a pass/fail promotion requirement.

The test case is curriculum order. Given two concepts, the audit constructs paired trajectories that share exactly the same learner prefix, instructional events, inserted outcomes, and delayed suffix but swap the early concept blocks. It counterbalances suffix order, holds probes fixed, and pairs effects within learner. The resulting contrast isolates order from exposure dose, item composition, realized responses, terminal recency, and probe-induced practice.

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

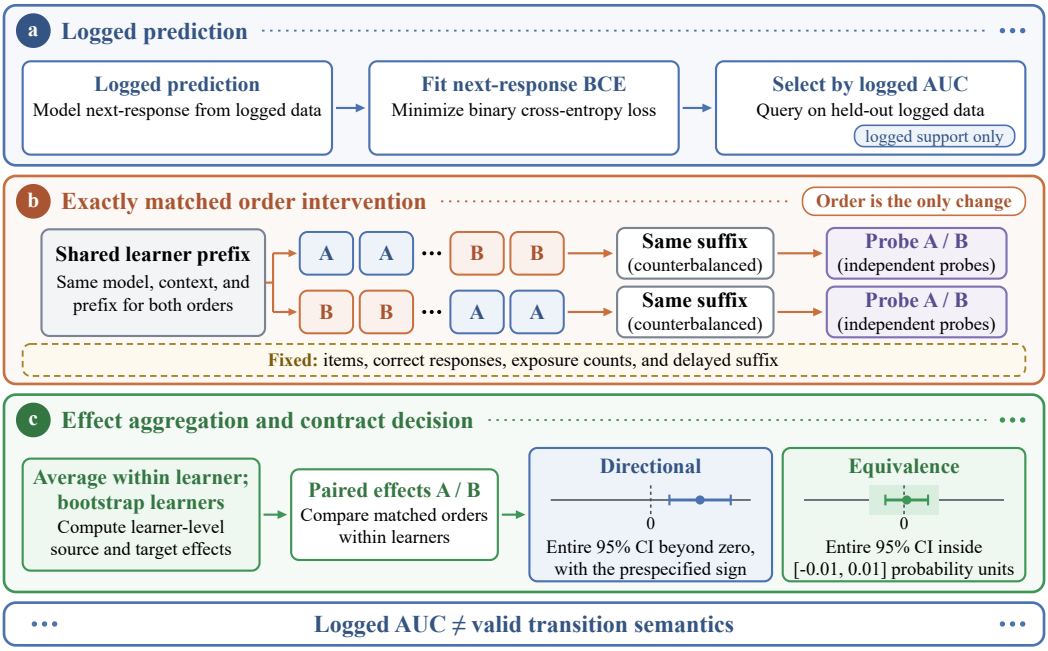

Figure 1: Process contracts add a second promotion axis beyond predictive fit. A model selected by logged next-response AUC is queried on paired curricula with the same learner prefix, item–response multiset, exposure dose, delayed suffix, and probes. Only early block order changes. Learner-level source and target effects must satisfy prespecified directional or equivalence contracts, so the audit tests transition semantics that logged AUC does not identify.

The audit has two roles. On public logs, it diagnoses model behavior but does not identify a causal effect in human learners. In a transparent simulator, the data-generating process supplies known directional, reversed, independent, and recency-only contracts. These contracts test whether models selected by predictive accuracy recover the intervention semantics of their environment.

This paper makes three contributions. First, it defines **process contracts** as a promotion criterion separate from predictive fit and shows that logged next-response risk does not identify the responses those contracts specify on controlled, off-support trajectories. Second, it makes each contract falsifiable through a **matched audit**. Exact multiset matching, suffix counterbalancing, independent probes, learner-level uncertainty, and strict signed and equivalence tests separate directional failure from spurious sensitivity and from mere uncertainty. On ASSISTments 2009, all directional-minus-neutral contrasts across five published architectures and three training seeds exclude zero, yet each architecture yields both signs within a 0.0063 AUC span. On a fresh 600-checkpoint matrix, contract satisfaction depends on architecture and logging policy even when test AUC stays high. DKT passes 24/24 endpoint contracts and SAKT passes 16/24. Four of the eight SAKT failures occur at a mean test AUC of at least 0.85. Third, it turns a failed contract into a training signal with **Process-Contrastive Regularization (PCR)**. PCR differentiates through the matched operator and is audited on identities outside its auxiliary loss. PCR reduces failed contracts from 4/16 to 1/16 in a frozen single-seed pilot and from 11/24 to 6/24 for SAKT across fresh seeds and three logging policies, without violating the predeclared predictive-fidelity guard. The same promotion gate refuses PCR trained against a deliberately misspecified contract.

What the paper adds. Knowledge-tracing evaluation scores a predictor against logged responses (Liu et al., 2022; Kim & Kim, 2026), and consistency methods constrain how predicted mastery moves when the correctness of a logged response is edited (Zhang et al., 2023). Neither specifies how a model must respond to a curriculum absent from the log. That response is what a planner queries. Process contracts supply that specification and make promotion an audit–repair–verify loop. Predictive likelihood still selects models on logged support, but a joint gate promotes a model only when process validity improves without an unacceptable loss of predictive fidelity.

2 PROCESS CONTRACTS AS A SECOND VALIDATION AXIS

2.1 LEARNER MODELS UNDER CONTROLLED TRAJECTORIES

Figure 1 summarizes the promotion from logged prediction to a contract-based process audit. For learner i, let h_i denote the logged history and $f_\theta(q \mid h)$ the model probability of a correct response to item q after history h. For source concept A and target concept B, two interventions $c_s^{A \prec B}$ and $c_s^{B \prec A}$ share the same item–response multiset, exposure dose, and delayed suffix but reverse the early A/B blocks. Index s denotes one of two counterbalanced suffix orders. Probes are scored independently so that one probe cannot teach the model before another is evaluated.

For endpoint concept $K \in \{A, B\}$ with probe items Q_K, the learner-level order effect is

$$\delta_{i,K} = \frac{1}{2} \sum_s \left[\frac{1}{|Q_K|} \sum_{q \in Q_K} f_\theta(q \mid h_i, c_s^{A \prec B}) - \frac{1}{|Q_K|} \sum_{q \in Q_K} f_\theta(q \mid h_i, c_s^{B \prec A}) \right]. \tag{1}$$

The audit averages effects within each learner before bootstrapping learners, the independent units. A directional endpoint passes only when its full 95% confidence interval lies on the expected side of zero. A null endpoint passes equivalence only when its full interval lies within $[-0.01, 0.01]$ probability units, the predeclared null band for calling an effect practically null (Riesthuis, 2024).

2.2 AUDIT INVARIANTS AND DECISION LOGIC

The order effect is informative only to the extent that the paired trajectories remove alternative explanations. Table 1 lists the invariants enforced by construction. A common learner prefix fixes the model state before intervention, and exact multiset matching fixes exposure dose and item composition. Fixed inserted responses prevent either arm from receiving an easier realized outcome path. Counterbalanced suffix order removes a privileged terminal concept, and independent probe copies prevent measurement of one endpoint from updating the state used to measure the other. The only remaining designed difference is therefore the order of the early A/B blocks.

Table 1: Matched-audit invariants and the alternative explanations they remove

Invariant	Removed variation
Common learner prefix	Pre-intervention model state and learner history
Same item–response multiset	Content, exposure dose, and inserted outcomes
Same delayed suffix	Terminal curriculum and reinforcement content
Counterbalanced suffix order	Mechanical last-concept recency
Independent probe copies	Practice induced by measuring another endpoint

Pairing continues through inference. Probes and suffix orders are first averaged within each learner, and the bootstrap then resamples learners rather than individual predictions. In the confirmation matrix, model seeds are averaged within each generated dataset before uncertainty is computed across dataset seeds, because checkpoints trained on one generated population are not additional populations. The hierarchy also makes the contract deliberately harder to satisfy than a point-estimate sign test.

Directional and null endpoints impose different interval criteria. A directional contract requires the entire interval to lie on the prespecified side of zero and thus to exclude both zero and the wrong sign. A null contract requires the entire interval to lie inside the predeclared null band, not merely to include zero. The band half-width of 0.01 is fixed before evaluation, so a wide interval crossing zero fails rather than counting as evidence of no order effect. Requiring the entire interval under both criteria means that uncertainty counts against contracts in null regimes, not only in transfer regimes.

The order effect is a controlled-response trajectory contrast. Curriculum order changes while inserted outcomes stay fixed and matched, so the contrast audits the state-transition semantics of the model under a reproducible intervention. It is not a policy-value estimand that marginalizes over learner-generated responses, nor does it measure the expected benefit of one curriculum over another.

Predictive risk does not identify the audit. Let μ be the logging distribution over queried histories, and suppose the controlled histories in an audit set $\mathcal{C}$ lie outside the support of μ. For any predictor f minimizing logged next-response risk, define $g = f$ on the support of μ and let g be arbitrary on $\mathcal{C}$. Then f and g have identical logged risk but can have arbitrarily different order effects $\delta_{i,K}$. Next-response optimality alone therefore cannot identify process validity. Identifying it requires overlap, inductive bias, or an additional process constraint. The experiments measure how large this underdetermination remains under a controlled simulator and practical logging policies.

2.3 What the audit certifies

Three claims are often conflated. *Predictive validity* concerns risk on histories sampled from the logging distribution. *Process validity* concerns the response of a model to a specified controlled trajectory transformation. *Human causal validity* concerns the effect of actually assigning a curriculum to students. The matched audit targets process validity. It can falsify a learner model as a simulator even when its test AUC is high, but it cannot by itself establish that the specified contract is a true causal law of human learning.

This distinction determines how results are used. On real logs, the audit is construct validation. It asks whether architectures that fit similar observations encode different transition semantics. In the simulator, the known state machine supplies ground-truth contracts, so failures identify a mismatch between a learned transition rule and its environment. For deployment, a contract should come from domain knowledge, a validated simulator, or prospective evidence. Passing then establishes consistency with that contract, not universal model correctness. Process validity is therefore modular and claim-specific. A model can pass curriculum-order contracts yet remain invalid for forgetting, hints, or long-horizon policy value.

2.4 Real-log controls and claim boundary

Concept relations are discovered from training learners only. Candidate pairs require bidirectional transition support and pure, globally disjoint item pools. A neutral control is admitted only when order balance, learner support, transition support, and per-item difficulty match the reference pair under predeclared calipers. Relation/design cells are matched before learner aggregation.

These controls characterize the response of a model to a trajectory permutation but do not turn observational logs into randomized curricula. The real-log claim therefore concerns disagreement and construct sensitivity of learned models, not prerequisite effects in students.

3 From audit to repair with process-contrastive regularization

3.1 Four-regime simulator

The synthetic environment defines four known contracts over two concepts. Their expected endpoint signs are $(0, +)$ for $A \to B$ transfer, $(-, 0)$ for $B \to A$ transfer, $(0, 0)$ for the independent regime, and $(0, 0)$ for the recency-only regime. The simulator records trajectories under uniform, blocked, and adaptive logging policies. The oracle must pass the audit before any learned-model result is interpreted.

3.2 Process-contrastive regularization

PCR adds a process loss to factual binary cross-entropy. For an $A \to B$ relation, the per-learner penalty is

$$\ell_{A\to B} = [|\delta_{i,A}| - \epsilon]_+ + [m - \delta_{i,B}]_+ , \tag{2}$$

where $m > 0$ is the directional margin and $\epsilon \geq 0$ is the null tolerance. The reverse regime swaps the directional endpoint and flips its sign, and the independent and recency-only regimes constrain both endpoints within the null tolerance. With ℓ_r the penalty of regime r and λ its weight, the complete objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{BCE}} + \lambda \, \mathbb{E}_i \left[\ell_r(\delta_{i,A}, \delta_{i,B}) \right] . \tag{3}$$

Figure 2 shows the leakage-safe training and promotion workflow.

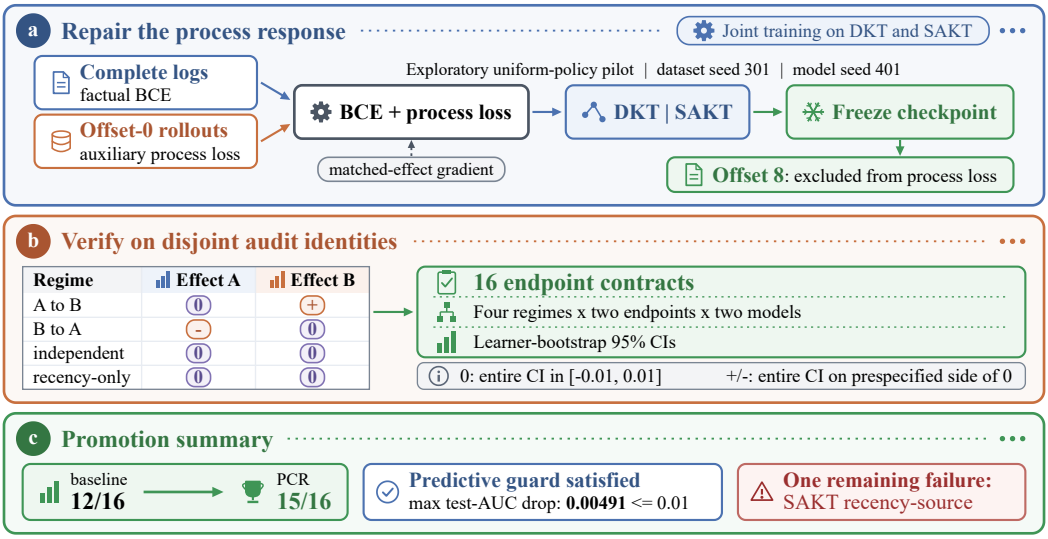

Figure 2: Leakage-safe audit–repair–verify workflow. Factual BCE preserves logged prediction, whereas PCR differentiates through matched offset-0 rollouts. The frozen audit evaluates the same endpoint contracts on disjoint offset-8 identities. Promotion requires fewer audit failures and no matched test-AUC drop above 0.01. PCR raises passed endpoint contracts from $12/16$ to $15/16$ in the single-seed repair pilot, where each interval over dataset seeds collapses to its point estimate.

PCR applies to any differentiable tracer and differentiates through the same matched rollout construction as the audit. Factual BCE is computed on the logged training batch, whereas only PCR uses the constructed matched rollouts.

Turning an audit into a training signal. PCR trains on learner-level effects rather than binary pass/fail labels. Before aggregation, it applies a hinge penalty that pushes directional endpoints past a margin and pulls null endpoints to within the null tolerance. The penalty preserves the geometry of the audit and supplies a gradient almost everywhere, even when no batch-level confidence interval is available during training. The same rollout operator defines the intervention at train and test time, but item identities differ. To improve, a model must therefore transfer across item identities rather than memorize the reported probes.

The factual and process terms constrain different objects, so promotion uses a joint gate rather than the weighted training loss alone. The factual BCE term protects ordinary response prediction on logged training data, whereas the process term constrains how the hidden state changes under a controlled permutation. A model can reduce either term without reducing the other. The gate therefore requires strictly fewer failed contracts on the audit identities and refuses any candidate whose worst matched AUC drop exceeds 0.01. It makes the repair claim auditable, since a process improvement cannot be bought by silently abandoning the predictor that motivates the model.

4 Experimental Protocol

Real-log audit. Five published model families (DKT, AKT, DKVMN, SAKT, and SimpleKT) are evaluated on ASSISTments 2009 and 2017 with three training seeds. Relation discovery and item selection use only training learners. Learner-level effects are computed on held-out learners, with learner-cluster bootstrap intervals.

Synthetic confirmation. The confirmation matrix crosses four regimes, three logging policies, five fresh dataset seeds, five fresh model seeds, and two published backbones (DKT and SAKT), for 600 checkpoints. Training uses length-64 trajectories and at most 20 epochs with patience 5. The excluded development run uses dataset/model seeds 0/42, and confirmation uses dataset seeds 101–105 and model seeds 201–205.

Repair protocol. Matched PCR and baseline runs use the same fresh 16-item-per-concept datasets. The auxiliary rollouts of PCR use offset 0, and audit evaluation uses offset 8. The auxiliary and audit item sets are disjoint. The exploratory repair pilot fixes $\lambda = 1$, $m = 0.01$, $\epsilon = 0.01$, dataset seed 301, and model seed 401 before any audit effect is evaluated. PCR is promoted only if it fails strictly fewer of the 16 endpoint contracts than its matched baseline and if no matched regime/model cell loses more than 0.01 in mean test AUC.

Replication hierarchy and frozen analysis. The three experimental layers stay separate because they answer different questions. The excluded seed-0/42 development run serves only to debug the simulator and form hypotheses. The 600-checkpoint confirmation matrix uses fresh dataset and model seeds and evaluates architecture–policy interactions without PCR. The seed-301/401 repair pilot asks a narrower feasibility question on identities outside the auxiliary loss, and a multi-seed repair confirmation repeats it (Appendix G.2). Development, confirmation, and repair estimates are never pooled. Within confirmation, five model seeds are averaged inside each of five dataset seeds. Percentile intervals use 2,000 resamples of the five dataset-seed means, so model-seed fits do not count as independent datasets. Appendix F.1 reports two sensitivities of these intervals.

Fail-closed evaluation. Machine-readable receipts record the expected regime, logging policy, model, seed, audit-design parameters, dataset and checkpoint hashes, and the complete result grid. Aggregate pass counts are recomputed from learner- or dataset-level rows rather than copied from training logs, because one missing regime, an overlapping item partition, or an incorrectly aggregated seed changes the meaning of a pass count.

5 Results

5.1 Predictive agreement masks process disagreement

In ASSISTments 2009, checkpoints with similar test AUC respond differently to matched order interventions. A strict neutral control admits two matched relations. On 304 common relation/design cells from 166 learners, the directional-minus-neutral confidence intervals of all five architectures at three training seeds exclude zero (Table 2). At seed 42, contrasts are positive for AKT, DKT, and DKVMN and negative for SAKT and SimpleKT. Across seeds, every architecture yields both signs while its test AUC varies by at most 0.0063, and SAKT moves from -0.01746 to $+0.04180$. ASSISTments 2017 has no admissible strict neutral control for at least one reference relation, which marks an applicability boundary rather than grounds for loosening the control criteria.

Table 2: ASSISTments 2009 directional-minus-neutral diagnostic at three training seeds, with test AUC in parentheses. Entries are learner-bootstrap means on matched cells (Appendix E.1). All 95% intervals exclude zero.

Model	Seed 42	Seed 43	Seed 44
AKT	$+0.00605$ (0.750)	-0.00135 (0.748)	-0.00052 (0.751)
DKT	$+0.00264$ (0.755)	$+0.01309$ (0.754)	-0.02882 (0.759)
DKVMN	$+0.00059$ (0.757)	$+0.00061$ (0.751)	-0.00036 (0.755)
SAKT	-0.01746 (0.743)	-0.01660 (0.739)	$+0.04180$ (0.744)
SimpleKT	-0.00319 (0.749)	$+0.00678$ (0.744)	$+0.00057$ (0.749)

5.2 High AUC does not determine contract satisfaction

In the excluded development run, the synthetic oracle and DKT pass every directional and null contract. SAKT fails the reversed directional endpoint and produces false recency order effects of $+0.0295$ and -0.0380 despite a null oracle. These development observations motivate the confirmation matrix but are not pooled with it.

The confirmation matrix (Figure 3) shows that the SAKT recency failures are not an oracle or single-seed artifact. They recur on fresh seeds, whereas the reversed-endpoint failure of the development run does not. The oracle and DKT pass all 24 endpoint contracts, whereas SAKT passes 16. Enumerating all 3125 bootstrap resamples changes no DKT decision and one SAKT decision, and Bon-

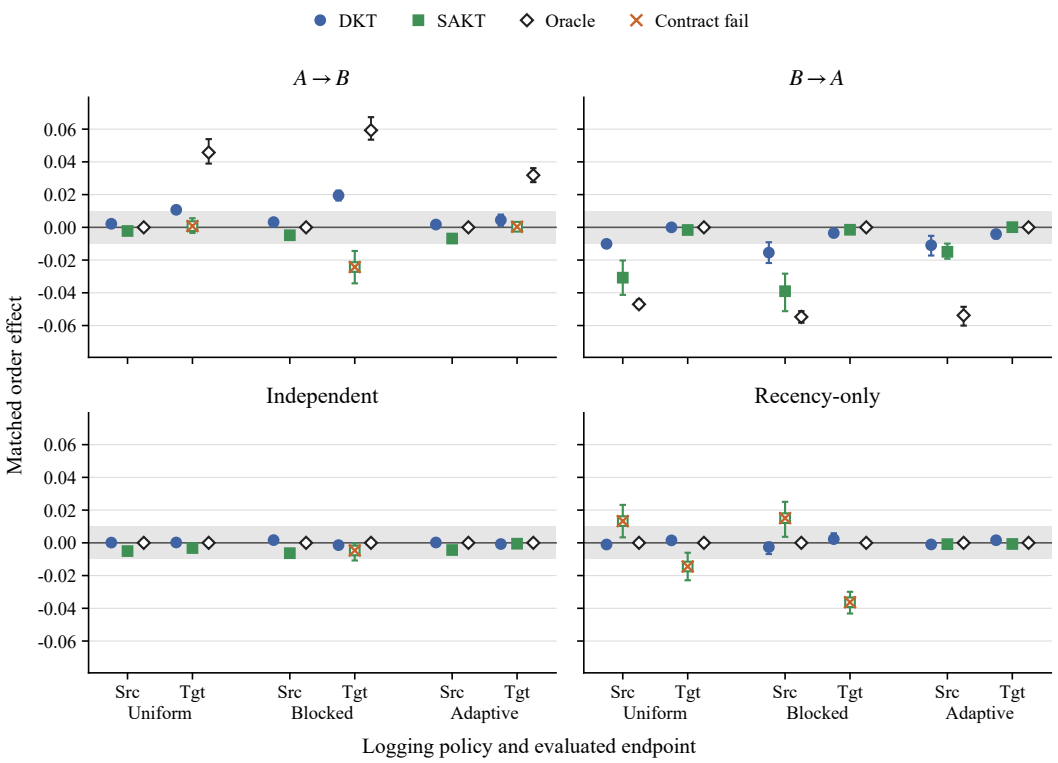

Figure 3: Fresh-seed matched order effects. Points average five model seeds within each of five independent datasets. The intervals bootstrap the five dataset-seed means. The gray band is the predeclared $[-0.01, 0.01]$ null band, and orange crosses mark failed directional or equivalence contracts. All oracle contracts pass. The crossed SAKT targets in the $A \to B$ panel and the crossed blocked-policy target in the Independent panel have a mean test AUC of at least 0.85.

ferroni leaves DKT at 23 and SAKT at 16 (Appendix F.1). A 0.005 band adds four failures for DKT and six for SAKT (Appendix F.2). Four of the eight failed SAKT endpoints still have a mean test AUC of at least 0.85, so failure is not confined to the low-AUC recency cells. For example, SAKT learns a significant target effect in the wrong direction under blocked $A \to B$ logging (-0.0243, 95% CI $[-0.0342, -0.0144]$, AUC 0.873). Under adaptive and uniform $A \to B$ logging, the same target effect attenuates until its interval includes zero, although mean AUC remains about 0.86.

Failures are qualitative, not a single calibration defect. The failed endpoints fall into three process-semantic classes. First, a model can reverse a directional response, as SAKT does for the blocked forward target. Second, an expected directional effect can attenuate until its interval includes zero, as the SAKT forward target does under adaptive and uniform logging. Third, a null endpoint can acquire an order response where the oracle effect is zero, as SAKT does in the recency-only regime and at the edge of the band in the independent regime. These heterogeneous failures suggest that scalar probability recalibration alone is unlikely to suffice. The required changes include reversing one contrast, amplifying another, and suppressing a third while preserving the matched structure. The endpoint-level results therefore show which transition semantics fail instead of compressing them into one score.

The distribution of failures across logging policies shows that predictive model selection alone leaves a process-semantic degree of freedom unresolved. The environment is unchanged, but the support and ordering statistics exposed during training differ. A model may satisfy a contract under uniform logging yet fail it under blocked or adaptive logging, even when test AUC remains similar. In the independent regime, SAKT fails the target null contract only under blocked logging, at test AUC 0.850. DKT, by contrast, passes every contract under all three policies. Planning creates the same kind of shift, because a new policy induces queries that depart from the correlations that make next-

response prediction easy. The result does not rank one architecture universally above another, and the same architecture trained under another logging policy needs its own audit.

5.3 PCR REPAIR TRANSFERS TO ITEM IDENTITIES OUTSIDE ITS AUXILIARY LOSS

PCR passes the frozen promotion gate. Matched baselines fail 4 of 16 contracts, whereas PCR fails 1 of 16. The contracts span four uniform-policy regimes, two published architectures, and two endpoints. All four baseline failures involve SAKT. PCR changes the forward-transfer target effect from -0.0679 to $+0.0462$, the reverse-transfer source effect from $+0.0006$ to -0.0147, and the recency target effect from -0.0254 to -0.0068. Each of these effects now satisfies its predeclared directional or $[-0.01, 0.01]$ null contract. The remaining failure is the SAKT recency source effect, which changes only from $+0.0374$ to $+0.0371$.

PCR also satisfies the predictive guard of the gate. The largest test-AUC drop across the eight matched regime/model cells is 0.00491 (SAKT in the recency-only regime) and lies below the frozen 0.01 guard. The next largest is 0.00130. PCR therefore passes both parts of the promotion gate, strictly fewer contract failures and no disallowed AUC drop. With one fresh dataset seed and one model seed, the pilot is an exploratory feasibility result, not a population estimate.

What the repair establishes. PCR repairs three SAKT errors: a wrong-sign transfer response, a missing reversed-transfer response, and a false recency target effect. The unchanged recency-source failure shows that the auxiliary loss does not force every endpoint to pass. Because the audit identities are absent from the auxiliary loss, the repair indicates transfer across items rather than direct probe fitting. Across fresh seeds and three logging policies, PCR repairs five of 11 SAKT failures at a worst matched AUC drop of 0.00525 (Appendix G.2). The promotion gate admits PCR at all three examined bands (Appendix F.2). Admission is relative to the matched baseline.

6 DISCUSSION

The selection object is a feasible set, not one score. The results show why promotion is underspecified if the learner model that wins on validation AUC is then treated as a simulator. Logged prediction measures agreement with observed responses, whereas planning needs agreement with controlled state transitions. Neither objective dominates the other. A model with process validity but unusable predictive fidelity is not a useful learner model. A highly predictive model without process validity can mis-rank curricula. The selection object is therefore a feasible set of models that meet a predictive guard and every contract for the intended policy class.

Process contracts make simulator assumptions executable. Every simulator embeds assumptions about transfer, independence, and recency. Ordinary training leaves those assumptions implicit in architecture and data, whereas the matched audit writes them as testable endpoint contracts. Writing them down does not eliminate judgment, since a wrong contract can regularize a model in the wrong direction. It still moves that judgment into an inspectable specification. Null contracts extend the audit from missing effects to spurious sensitivity. The four-regime simulator shows both sides. The same operator requires a signed response in one regime and invariance in another.

Audit–repair–verify is a reusable promotion protocol. For a new use case, practitioners can specify high-consequence transformations, build matched trajectories, and run an oracle or domain-consistency gate before comparing models. A model that fails may remain a useful predictor, but it is an unsupported simulator for that intervention family. When a contract is justified, PCR can precede an audit on item identities outside its auxiliary loss and the original predictive guard. The protocol localizes failure without treating one synthetic benchmark as general certification.

7 RELATED WORK

Knowledge-tracing consistency and counterfactual tests. Prior knowledge-tracing research has improved next-response prediction, introduced causal or graph-structured inductive biases, constrained monotonic responses to correctness edits, and used counterfactual histories to explain predictions or recommend instruction (Kumar et al., 2023; Hong et al., 2025; Zhang et al., 2023; Cui

et al., 2024; Kim et al., 2026). These methods mainly modify response predictions, explain them, or recommend instruction from them. Process contracts instead test prespecified transition semantics before a predictor is promoted to a simulator. Recent circuit analysis shows that transformer knowledge tracers can adopt permutation-invariant shortcuts on hierarchical tasks (Liu et al., 2026), an order insensitivity that directional contracts expose.

Knowledge-tracing evaluation practice. The field standardizes predictor comparison through a library that fixes preprocessing and splits, through a large benchmark corpus, and through re-evaluations that report how much of a published gain survives a controlled protocol (Liu et al., 2022; 2023b; Sarsa et al., 2022; Kim & Kim, 2026). These instruments score predictors against logged responses, and none tests how a model changes its own state under an intervention absent from the log. That state change is what a planner reads and what a process contract specifies.

World-model evaluation and educational planning. The broader world-model literature separates short-horizon prediction from procedural planning and identifies train–test mismatch when next-state predictors are optimized over unseen action sequences (Chen et al., 2025; Parthasarathy et al., 2025). Recent educational systems already use dynamic cognitive diagnosis for learning-path planning and explicit state-transition simulators for virtual schools (Sheng et al., 2026; Ye et al., 2026). Evaluation protocols for such simulators score behavioral fidelity and responsiveness to guidance (Yang et al., 2026). Both criteria are read against the observed responses of a learner rather than against prespecified transition semantics. The validity of simulated learners remains debated. Learner simulation has been defined with linguistic, behavioral, and cognitive metrics under which prompting and fine-tuning alike fall short (Scarlatos et al., 2026). Epistemic fidelity rather than surface realism has been argued to be the property worth demanding (Yuan et al., 2026). These lines make intervention validity consequential but leave open an educationally grounded promotion test. This paper contributes a contract-based promotion test that combines matched transformations, strict decisions, and repair verified on audit identities outside the auxiliary loss.

8 Limitations

The synthetic contracts are deliberately small and do not establish that the same structure governs human learning. The real-log analysis is observational. Matched controls reduce specific confounds of composition and support but cannot identify a causal curriculum effect, and some datasets and relations lack strict controls. Because PCR teaches a specified process prior, a wrong prior can make a model confidently wrong. At the predeclared null band, the promotion gate refuses PCR trained against a misspecified contract only because the audit scores the true contract (Appendix G.3). The repair result therefore establishes feasibility under a contract assumed correct, not identification of the contract. The single-seed pilot decides its contracts on point estimates of one dataset, so the multi-seed repair confirmation (Appendix G.2) carries the promotion claim. Short binary-response trajectories over two concepts do not cover richer tutoring processes such as forgetting, hints, open-ended feedback, or multi-session transfer.

9 Conclusion

Process contracts provide an executable second validation axis for promoting a knowledge tracer from predictor to simulator. Exact-multiset curriculum pairs isolate controlled transition semantics. Directional and equivalence contracts make missing and spurious responses falsifiable. Uncertainty computed over learners and dataset seeds prevents probe and checkpoint multiplicity from inflating evidence. On ASSISTments 2009, every architecture yields both signs across training seeds while its test AUC varies by at most 0.0063. The confirmation matrix shows that high AUC does not determine transition semantics. Contract satisfaction depends on the architecture and the logging policy behind the training data. PCR repairs three of four failures in the single-seed pilot and five of 11 for SAKT across fresh seeds and three logging policies, within a predeclared predictive guard. The promotion gate refuses PCR trained against a misspecified contract. A repair counts only when it fails fewer contracts than its baseline on item identities outside the auxiliary loss. Unlike a predictive benchmark, the matched audit can reject a high-AUC model. Process validity should therefore be tested, not assumed, whenever a predictive learner model is reused for planning.

AI USE STATEMENT

The authors use generative AI tools to help develop the process-contract framework, propose and refine the hypotheses, design the methodology and the experiments, and interpret the results. The tools also assist in implementing the methods and the experiment code, including the simulator that generates the synthetic datasets, and help search the literature, write the figure code, and draft and edit the manuscript. Language-model reviews of the manuscript give feedback on the methodology, and the multi-seed confirmation and the misspecified-contract ablation of Appendices G.2 and G.3 respond to that feedback. The paper states no theorem and analyzes no qualitative data, so the disclosure tasks concerning mathematical claims, proofs, and qualitative or thematic analysis do not apply. The authors review all AI-assisted work. Cited sources are checked against primary records; the code for the audit, the simulator, and PCR is unit tested; and a verification script recomputes the main results from machine-readable receipts. The authors take responsibility for the content of this work, including its text, claims, and code.

ETHICS STATEMENT

This work analyzes public educational interaction datasets and synthetic learners, makes no causal claims about individual students, and deploys no instructional policies. A practical risk is treating an audited or regularized simulator as evidence that an intervention benefits real learners. The paper therefore separates process validity from human causal validity and recommends prospective educational evaluation before deployment.

REFERENCES

Ghodai Abdelrahman, Qing Wang, and Bernardo Pereira Nunes. Knowledge tracing: A survey. *ACM Computing Surveys*, 55(11):224:1–224:37, 2023. doi: 10.1145/3569576. URL https://doi.org/10.1145/3569576.

Delong Chen, Willy Chung, Yejin Bang, Ziwei Ji, and Pascale Fung. Worldprediction: A benchmark for high-level world modeling and long-horizon procedural planning. *arXiv preprint arXiv:2506.04363*, 2025. doi: 10.48550/arXiv.2506.04363. URL https://arxiv.org/abs/2506.04363.

Jiajun Cui, Minghe Yu, Bo Jiang, Aimin Zhou, Jianyong Wang, and Wei Zhang. Interpretable knowledge tracing via response influence-based counterfactual reasoning. In *2024 IEEE 40th International Conference on Data Engineering*, pp. 1103–1116. IEEE, 2024. doi: 10.1109/ICDE60146.2024.00090. URL https://doi.org/10.1109/ICDE60146.2024.00090.

Aritra Ghosh, Neil T. Heffernan, and Andrew S. Lan. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2330–2339. ACM, 2020. doi: 10.1145/3394486.3403282. URL https://doi.org/10.1145/3394486.3403282.

Yihong Gu, Cong Fang, Yang Xu, Zijian Guo, and Jianqing Fan. Fundamental computational limits in pursuing invariant causal prediction and invariance-guided regularization. *arXiv preprint arXiv:2501.17354*, 2025.

Kevin Hong, Kia Karbasi, and Gregory Pottie. Extracting causal relations in deep knowledge tracing. In *Proceedings of the 18th International Conference on Educational Data Mining*, pp. 586–591, Palermo, Italy, July 2025. International Educational Data Mining Society. doi: 10.5281/zenodo.15870217. URL https://educationaldatamining.org/EDM2025/proceedings/2025.EDM.poster-demo-papers.266/index.html.

Jaewook Kim and Hyeoncheol Kim. Ensuring reliability in programming knowledge tracing: A re-evaluation of attention-augmented models and experimental protocols. In *Proceedings of the International Conference on Intelligent Tutoring Systems (ITS)*, 2026. arXiv:2605.04727.

Woojin Kim, Changkwon Lee, and Hyeoncheol Kim. Ktcf: Actionable recourse in knowledge tracing via counterfactual explanations for education. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(45):38726–38735, 2026. doi: 10.1609/aaai.v40i45.41216. URL https://doi.org/10.1609/aaai.v40i45.41216.

Nischal Ashok Kumar, Wanyong Feng, Jaewook Lee, Hunter McNichols, Aritra Ghosh, and Andrew Lan. A conceptual model for end-to-end causal discovery in knowledge tracing. In *Proceedings of the 16th International Conference on Educational Data Mining*, pp. 413–418, Bengaluru, India, July 2023. International Educational Data Mining Society. doi: 10.5281/zenodo.8115727. URL https://educationaldatamining.org/EDM2023/proceedings/2023.EDM-posters.42/index.html.

Nathan Lambert, Kristofer Pister, and Roberto Calandra. Investigating compounding prediction errors in learned dynamics models. *arXiv preprint arXiv:2203.09637*, 2022.

Naiming Liu, Richard Baraniuk, and Shashank Sonkar. Circuit complexity of hierarchical knowledge tracing and implications for log-precision transformers, 2026. URL https://arxiv.org/abs/2603.23823.

Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, Jiliang Tang, and Weiqi Luo. pyKT: A python library to benchmark deep learning based knowledge tracing models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1347. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/75ca2b23d9794f02a92449af65a57556-Abstract-Datasets_and_Benchmarks.html.

Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, and Weiqi Luo. simpleKT: A simple but tough-to-beat baseline for knowledge tracing. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=9HiGqC9C-KA.

Zitao Liu, Qiongqiong Liu, Teng Guo, Jiahao Chen, Shuyan Huang, Xiangyu Zhao, Jiliang Tang, Weiqi Luo, and Jian Weng. XES3G5M: A knowledge tracing benchmark dataset with auxiliary information. In *Advances in Neural Information Processing Systems*, volume 36, 2023b. doi: 10.52202/075280-1429. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/67fc628f17c2ad53621fb961c6bafcaf-Abstract-Datasets_and_Benchmarks.html.

Yotam Norman and Ron Meir. Unsupervised representation learning – an invariant risk minimization perspective. *arXiv preprint arXiv:2505.12506*, 2025.

Shalini Pandey and George Karypis. A self attentive model for knowledge tracing. In *Proceedings of the 12th International Conference on Educational Data Mining*, pp. 384–389. International Educational Data Mining Society, 2019. URL https://arxiv.org/abs/1907.06837.

Arjun Parthasarathy, Nimit Kalra, Rohun Agrawal, Yann LeCun, Oumayma Bounou, Pavel Izmailov, and Micah Goldblum. Closing the train-test gap in world models for gradient-based planning, 2025. URL https://arxiv.org/abs/2512.09929.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J. Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In *Advances in Neural Information Processing Systems 28*, pp. 505–513, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/bac9162b47c56fc8a4d2a519803d51b3-Abstract.html.

Paul Riesthuis. Simulation-based power analyses for the smallest effect size of interest: A confidence-interval approach for minimum-effect and equivalence testing. *Advances in Methods and Practices in Psychological Science*, 7(2):25152459241240722, 2024. doi: 10.1177/25152459241240722.

Sami Sarsa, Juho Leinonen, and Arto Hellas. Empirical evaluation of deep learning models for knowledge tracing: Of hyperparameters and metrics on performance and replicability. *Journal of Educational Data Mining*, 14(2):32–102, 2022. doi: 10.5281/zenodo.7086179. URL https://doi.org/10.5281/zenodo.7086179.

Alexander Scarlatos, Jaewook Lee, Simon Woodhead, and Andrew Lan. Simulated students in tutoring dialogues: Substance or illusion? In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, 2026. arXiv:2601.04025.

Ziji Sheng, Guiyao Tie, Weidong Wang, Pan Zhou, and Daizong Liu. LearnerCoMPASS: Intelligent tutoring system with dynamic cognitive diagnosis and multi-model path planning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9016–9042. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.408. URL https://aclanthology.org/2026.acl-long.408/.

Claas A. Voelcker, Anastasiia Pedan, Arash Ahmadian, Romina Abachi, Igor Gilitschenski, and Amir-Massoud Farahmand. Calibrated value-aware model learning with probabilistic environment models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 61745–61768. PMLR, 2025. URL https://proceedings.mlr.press/v267/voelcker25a.html.

Ran Wei, Nathan Lambert, Anthony D. McDonald, Alfredo García, and Roberto Calandra. A unified view on solving objective mismatch in model-based reinforcement learning. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=tQVZgvXhZb. arXiv:2310.06253.

Ke Yang, Chenglong Wang, Michel Galley, Chandan Singh, Jeevana Priya Inala, ChengXiang Zhai, and Jianfeng Gao. Studentsim: Training llm-based student simulators. *arXiv preprint arXiv:2609.01591*, 2026.

Yulei Ye, Wenhao Li, Zhong Wen, Yunshu Huang, Yichen Hu, Zifan Wei, Yige Wang, Xinyu Xie, Haoxuan Yang, Yanjun Huang, Ruijia Li, Hong Qian, Yu Song, Bo Jiang, Bingdong Li, Lijun Li, Bo Zhang, Pinlong Cai, Xingcheng Xu, Shuangye Chen, Xia Hu, Liang He, Aimin Zhou, Jingjing Qu, Jing Shao, and Xiangfeng Wang. Agentschool: An llm-powered multi-agent simulation for education, 2026. URL https://arxiv.org/abs/2605.30144.

Zhihao Yuan, Yunze Xiao, Ming Li, Weihao Xuan, Richard Tong, Mona Diab, and Tom Mitchell. Towards valid student simulation with large language models. *arXiv preprint arXiv:2601.05473*, 2026.

Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 765–774. International World Wide Web Conferences Steering Committee, 2017. doi: 10.1145/3038912.3052580. URL https://doi.org/10.1145/3038912.3052580.

Moyu Zhang, Xinning Zhu, Chunhong Zhang, Wenchen Qian, Feng Pan, and Hui Zhao. Counterfactual monotonic knowledge tracing for assessing students' dynamic mastery of knowledge concepts. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 3236–3246. Association for Computing Machinery, 2023. doi: 10.1145/3583780.3614827. URL https://doi.org/10.1145/3583780.3614827.

REPRODUCIBILITY STATEMENT

The appendix records splits, interventions, matching, simulator equations, hyperparameters, and seeds. The code, to be released with the paper, verifies configuration and artifact hashes and regenerates the reported statistics, from learner-level rows for the real-log audit and from dataset-seed means for the confirmation matrix.

A AUDIT AND MATCHING DETAILS

The synthetic audit uses the first $t_0 = 20$ interactions of each held-out learner as a common prefix. Each intervention contains four two-event blocks, and all inserted responses equal one. The paired curricula swap the two early concept blocks while holding the reinforcement blocks fixed, so both curricula match in item–response multiset, exposure dose, and length. Both reinforcement-block orders are evaluated. Two additional items measure each source and target endpoint. Every probe is appended to an independent copy of the history, so the first measurement cannot become practice for the second.

Within each suffix order, the matched audit subtracts target-first predictions from source-first predictions. For synthetic data, the two suffix-order effects are averaged within learner before a nonparametric learner bootstrap with 2,000 resamples in each dataset. Directional contracts require the entire 95% interval to have the expected strict sign, and null contracts require the interval to lie inside $[-0.01, 0.01]$. In the confirmation matrix, model-seed estimates are averaged within each independently generated dataset before uncertainty is bootstrapped over the five dataset seeds. This hierarchy avoids treating multiple fits to one generated dataset as independent datasets. The reported real-log control instead uses $t_0 = 50$, four probes per endpoint, and 10,000 learner-bootstrap resamples for the directional-minus-neutral contrast.

Evaluation receipts record dataset SHA-256 digests, split configuration, model identity and seed, checkpoint/configuration/metric digests, expected grids, and audit-design parameters. PCR receipts also record offset-0 PCR items and offset-8 evaluation items. These records establish disjoint configured identities for the repair audit. Evaluation derives within-dataset summaries from learner- or dataset-level rows. Aggregation recomputes cross-dataset intervals and contracts from dataset-level means independently of stored pass flags.

B Synthetic data-generating processes

Each learner begins with mastery $x_A = x_B = 0$. Outside the recency-only regime, the probability of a correct response on concept K is

$$p(y_t = 1 \mid K_t = K) = \sigma(-2 + 4x_K). \tag{4}$$

A correct response updates only the practiced concept,

$$x_K \leftarrow x_K + 0.5\, g_K(1 - x_K). \tag{5}$$

Incorrect responses do not update mastery. For $A \to B$, $g_A = 1$ and $g_B = 0.05 + 0.95x_A$, and $B \to A$ reverses the roles. In the independent regime, $g_A = g_B = 1$. In the recency-only regime, mastery never updates and the response logit is 2 when the current concept matches the immediately preceding concept and -2 otherwise.

Uniform logging selects either concept with probability $1/2$. Blocked logging randomly chooses an orientation for each learner and selects the concept prescribed for the current half by that orientation with probability 0.8. With probability 0.8, adaptive logging selects the concept with lower current oracle success probability. Item identities are drawn uniformly within the selected concept. The confirmation matrix uses eight items per concept. Both repair experiments use 16 items per concept so that the six PCR training identities (offset 0) and six audit identities (offset 8) per concept are disjoint.

Before learned models are interpreted, the same audit runs on the exact state-machine oracle. The oracle gate requires every oracle endpoint of each cross-dataset aggregate to pass its prescribed directional or equivalence contract. There are 24 such endpoints in the confirmation matrix (four regimes, three policies, and two endpoints) and eight in each repair aggregate. Aggregation receipts record that the gate passes for every aggregate. The unit tests also check simulator/oracle parity on forced action sequences and common random numbers.

C Receipts and released artifacts

Machine-readable receipts contain the complete regime–policy–dataset-seed–model-seed grid for the two evaluated backbones, test AUC values, dataset-seed effect means, bootstrap configuration, semantic configuration fingerprints, and artifact digests. These receipts and the predeclared PCR promotion decision are to be released with the code. Exploratory repair results are labeled separately from the confirmation matrix and are not pooled with it.

D Formal contract specification

The matched audit comprises a finite collection of endpoint process contracts, which it evaluates individually rather than collapsing into a single omnibus score. Let $I_K = [L_K, U_K]$ denote the

95% bootstrap confidence interval for the matched order effect on endpoint $K \in \{A, B\}$. The band half-width is $\tau = 0.01$, which equals the null tolerance ϵ used by PCR but is varied independently in Appendix F.2. The three primitive decisions are:

$$\mathrm{Pass}_+(I_K) \iff L_K > 0, \tag{6}$$

$$\mathrm{Pass}_-(I_K) \iff U_K < 0, \tag{7}$$

$$\mathrm{Pass}_0(I_K) \iff -\tau \le L_K \ \wedge \ U_K \le \tau. \tag{8}$$

The directional rules require the complete interval to lie on the prespecified side of zero, whereas the null rule requires it to lie within the closed null band. Thus, a wide interval that merely contains zero is inconclusive and fails the null contract. This containment logic follows the principle of prespecified equivalence bounds, under which nonsignificance is not evidence of no meaningful effect (Riesthuis, 2024). The 95%-interval containment rule is an operationally conservative contract, not a claim that the audit is identical to a classical 90%-interval TOST procedure. Table 3 gives the contract each endpoint must satisfy. The promotion gate requires strictly fewer failures than the matched baseline under the predictive guard, whereas the feasible set of Section 6 requires every contract.

Table 3: Complete endpoint contract matrix. Signs refer to $\delta_K = p_K(A \prec B) - p_K(B \prec A)$.

Data-generating regime	Source endpoint A	Target endpoint B
$A \to B$ transfer	$\mathrm{Pass}_0(I_A)$	$\mathrm{Pass}_+(I_B)$
$B \to A$ transfer	$\mathrm{Pass}_-(I_A)$	$\mathrm{Pass}_0(I_B)$
Independent learning	$\mathrm{Pass}_0(I_A)$	$\mathrm{Pass}_0(I_B)$
Recency-only response	$\mathrm{Pass}_0(I_A)$	$\mathrm{Pass}_0(I_B)$

The contract matrix makes two asymmetries explicit. First, the source endpoint is not expected to improve in the forward-transfer regime merely because the source concept appears first. Second, the sign in the reverse-transfer regime is negative because the estimand always subtracts the $B \prec A$ arm from the $A \prec B$ arm. This orientation is fixed once to avoid silent sign changes between regimes. Independent and recency-only regimes are both null under the counterbalanced estimand despite different response mechanisms.

D.1 Why both directional and null contracts are needed

Directional and null contracts jointly rule out two degenerate solutions. An audit that uses only directional contracts detects missing or reversed transfer but not order sensitivity invented where the environment has none. An audit that uses only null contracts passes a model that suppresses all order effects and thereby erases transfer that the environment contains. The PCR loss mirrors this structure. Its directional hinge pushes one endpoint past a margin, whereas its tolerance hinge pulls a null endpoint to within the null tolerance. Pass/fail decisions are made only after training and never serve as per-example labels.

E Matched trajectory construction and admissibility

The matched audit constructs two arms for every held-out learner. The construction is deterministic given the learner prefix, selected item identities, and suffix-order index.

1. Initialize two independent model states with the first $t_0 = 20$ logged interactions, before either state consumes an audit event.
2. For each concept, select two early items, two delayed-suffix items, and two probe items so that the three roles use distinct identities.
3. Append the early blocks as A, A, B, B in the first arm and B, B, A, A in the second, assigning the same response value to every pair of corresponding events.
4. Append the same two reinforcement blocks to both arms, and repeat the arm pair under both suffix orders (A-suffix then B-suffix, and the reverse).

5. For each endpoint, copy each completed history before appending each probe. Within each learner, average first across probe predictions and then across suffix orders.
6. Bootstrap over learners. In the confirmation matrix, average the five model seeds within each dataset seed, then bootstrap the five dataset-seed means.

The arms thus share the prefix, item–response multiset, exposure dose, suffix content, and probe identities. Only the early block order differs. Learner-level aggregation prevents probes and suffix replicates from counting as independent learners.

E.1 Real-log control selection

Because concept relations and item pools are estimated from observational logs, the real-log analysis adds an admissibility layer. Relation discovery uses only training learners. Each candidate reference pair requires bidirectional transition support and pure, globally disjoint item pools. Each neutral control meets the same structural requirements and is selected using learner support, forward and reverse transition counts, item-pool size, and item difficulty. Before learner aggregation, the final diagnostic intersects relation/design cells and excludes unsupported cells from both arms. Table 4 records the resulting reference/control map for ASSISTments 2009.

Table 4: Strict reference/control map for the cell-matched ASSISTments 2009 diagnostic, shared by the three training seeds. Counts and matching costs come from the frozen control receipt.

Ref. src.	Ref. tgt.	Ctrl. src.	Ctrl. tgt.	Ref. users	Ctrl. users	Ref. fwd./rev.	Ctrl. fwd./rev.	Difficulty cost
12	13	8	7	220	144	210/165	197/191	0.01074
12	59	15	53	184	201	71/44	40/28	0.00338

After intersection, the two accepted ASSISTments 2009 mappings yield 304 common relation/design cells across 166 learners. In ASSISTments 2017, at least one reference relation has no neutral control satisfying every strict criterion. The analysis therefore reports an applicability boundary instead of substituting a looser control after inspecting model effects.

Table 5: Directional-minus-neutral contrasts on ASSISTments 2009 for the five architectures at three training seeds. Entries are learner-bootstrap means and 95% intervals on 304 common relation/design cells from 166 learners.

Model	Seed	Test AUC	Difference (95% CI)
AKT	42	0.750	$+0.00605\ [0.00570, 0.00638]$
	43	0.748	$-0.00135\ [-0.00159, -0.00112]$
	44	0.751	$-0.00052\ [-0.00074, -0.00030]$
DKT	42	0.755	$+0.00264\ [0.00207, 0.00321]$
	43	0.754	$+0.01309\ [0.01206, 0.01414]$
	44	0.759	$-0.02882\ [-0.03023, -0.02743]$
DKVMN	42	0.757	$+0.00059\ [0.00057, 0.00061]$
	43	0.751	$+0.00061\ [0.00048, 0.00074]$
	44	0.755	$-0.00036\ [-0.00038, -0.00035]$
SAKT	42	0.743	$-0.01746\ [-0.01841, -0.01657]$
	43	0.739	$-0.01660\ [-0.01848, -0.01475]$
	44	0.744	$+0.04180\ [0.03998, 0.04361]$
SimpleKT	42	0.749	$-0.00319\ [-0.00366, -0.00271]$
	43	0.744	$+0.00678\ [0.00554, 0.00804]$
	44	0.749	$+0.00057\ [0.00042, 0.00072]$

The seed-43 and seed-44 checkpoints of the five architectures share the split, relations, control map, item designs, and settings of the seed-42 diagnostic. Rerunning the seed-42 control reproduces its receipt exactly. Table 5 reports all 15 contrasts. Each interval comes from a learner bootstrap within one checkpoint. Every interval excludes zero, every architecture yields both signs across seeds, and the descriptive Spearman correlation between test AUC and the contrast is 0.07 over the 15 checkpoints.

F COMPLETE FRESH-SEED CONFIRMATION MATRIX

Table 6 reports every endpoint for both evaluated backbones in the frozen confirmation aggregate. It covers four regimes, three logging policies, and two endpoints per architecture. All synthetic learned-model counts in this paper are restricted to the two published backbones, DKT and SAKT. Each interval is computed from five dataset-seed means that each average five model seeds. A bold superscript F marks a failed endpoint contract, whereas unmarked endpoints pass. Test AUC repeats across both endpoint columns of a regime–policy–model cell because test AUC is a checkpoint-level predictive metric, not an endpoint-specific audit statistic.

Table 6: Complete 48-endpoint synthetic confirmation matrix. Entries are mean [95% CI]; F marks contract failure.

Model	Regime	Policy	Source effect	Target effect	AUC
DKT	$A \to B$	Adaptive	+0.0017 [+0.0003, +0.0031]	+0.0044 [+0.0009, +0.0078]	0.863
DKT	$A \to B$	Blocked	+0.0032 [+0.0019, +0.0045]	+0.0195 [+0.0164, +0.0224]	0.876
DKT	$A \to B$	Uniform	+0.0022 [+0.0011, +0.0032]	+0.0107 [+0.0087, +0.0128]	0.864
DKT	$B \to A$	Adaptive	-0.0109 [-0.0172, -0.0052]	-0.0041 [-0.0067, -0.0018]	0.871
DKT	$B \to A$	Blocked	-0.0154 [-0.0218, -0.0091]	-0.0035 [-0.0051, -0.0022]	0.873
DKT	$B \to A$	Uniform	-0.0101 [-0.0120, -0.0086]	+0.0000 [-0.0010, +0.0009]	0.870
DKT	Independent	Adaptive	+0.0002 [-0.0005, +0.0010]	-0.0007 [-0.0018, +0.0000]	0.853
DKT	Independent	Blocked	+0.0017 [+0.0006, +0.0027]	-0.0014 [-0.0029, -0.0001]	0.856
DKT	Independent	Uniform	+0.0002 [-0.0006, +0.0010]	+0.0002 [-0.0010, +0.0013]	0.846
DKT	Recency	Adaptive	-0.0010 [-0.0014, -0.0007]	+0.0016 [+0.0008, +0.0025]	0.812
DKT	Recency	Blocked	-0.0026 [-0.0068, +0.0001]	+0.0023 [+0.0000, +0.0059]	0.863
DKT	Recency	Uniform	-0.0010 [-0.0023, -0.0001]	+0.0015 [+0.0006, +0.0027]	0.878
SAKT	$A \to B$	Adaptive	-0.0068 [-0.0082, -0.0055]	+0.0004 [-0.0013, +0.0022]^{F}	0.863
SAKT	$A \to B$	Blocked	-0.0048 [-0.0065, -0.0029]	-0.0243 [-0.0342, -0.0144]^{F}	0.873
SAKT	$A \to B$	Uniform	-0.0022 [-0.0030, -0.0015]	+0.0007 [-0.0034, +0.0056]^{F}	0.861
SAKT	$B \to A$	Adaptive	-0.0149 [-0.0192, -0.0099]	+0.0001 [-0.0007, +0.0008]	0.870
SAKT	$B \to A$	Blocked	-0.0390 [-0.0512, -0.0283]	-0.0015 [-0.0032, +0.0002]	0.870
SAKT	$B \to A$	Uniform	-0.0308 [-0.0413, -0.0203]	-0.0016 [-0.0034, +0.0001]	0.865
SAKT	Independent	Adaptive	-0.0044 [-0.0056, -0.0031]	-0.0005 [-0.0013, +0.0004]	0.854
SAKT	Independent	Blocked	-0.0063 [-0.0074, -0.0053]	-0.0047 [-0.0108, -0.0002]^{F}	0.850
SAKT	Independent	Uniform	-0.0050 [-0.0063, -0.0041]	-0.0032 [-0.0056, -0.0012]	0.842
SAKT	Recency	Adaptive	-0.0008 [-0.0011, -0.0006]	-0.0007 [-0.0014, +0.0000]	0.510
SAKT	Recency	Blocked	+0.0151 [+0.0037, +0.0250]^{F}	-0.0363 [-0.0432, -0.0299]^{F}	0.734
SAKT	Recency	Uniform	+0.0133 [+0.0034, +0.0232]^{F}	-0.0145 [-0.0229, -0.0060]^{F}	0.645

The table shows why headline pass counts are not interchangeable with a mean absolute effect, which cannot tell a wrong-sign response from a correct one. SAKT produces wrong-sign and attenuated directional responses as well as spurious null-regime responses. Its eight failures are the three $A \to B$ targets, the independent target under blocked logging, and the four recency endpoints under blocked and uniform logging. The $A \to B$ targets reverse under blocked logging and attenuate to intervals containing zero under adaptive and uniform logging. The first four failures occur at test AUC between 0.850 and 0.873, whereas the recency failures coincide with test AUC of 0.734 under blocked and 0.645 under uniform logging. DKT happens to satisfy this contract suite, but the audit does not certify DKT for untested transformations or human causal effects.

F.1 ENDPOINT-FAMILY AND BOOTSTRAP SENSITIVITY

Two sensitivity analyses test whether the pass/fail pattern depends on the predeclared per-endpoint 95% percentile bootstrap, which uses 2,000 resamples over the five dataset-seed means. The first enumerates all $5^5 = 3125$ bootstrap resamples of these means, eliminating resampling noise. The second retains exact enumeration and applies a Bonferroni correction at confidence $1 - 0.05/24$ over the 24 endpoints of each architecture, so the family-wise error rate is controlled at 0.05.

The separation between architectures is stable across both sensitivity analyses, although the pass/fail decisions for individual endpoints near a decision boundary are not stable. Table 7 reports the results. Under exact enumeration, DKT still passes 24 of 24, and SAKT passes 17 of 24. The pass/fail decision for the blocked-policy independent target of SAKT changes from fail to pass because the interval tightens from $[-0.01077, -0.00020]$ to $[-0.00965, -0.00020]$, inside the $[-0.01, 0.01]$ band. Only this decision depends on resampling noise. Under exact enumeration, three of the seven SAKT failures occur at a mean test AUC of at least 0.85, whereas four of eight do so under the predeclared

Table 7: Endpoint contracts passed out of 24 under the predeclared analysis and two sensitivities

Model	Predeclared 95%	Exact enumeration	Bonferroni over 24
DKT	24	24	23
SAKT	16	17	16

bootstrap. Under the Bonferroni correction, DKT passes 23 of 24, and SAKT passes 16 of 24. DKT fails the adaptive $A \to B$ target because the Bonferroni interval starts at -0.00078, whereas the exact 95% interval starts at $+0.00111$. Four of the 48 evaluated endpoints are within 0.002 of a decision boundary.

F.2 Equivalence-band sensitivity

Every contract is recomputed at a tighter band of half-width 0.005 and a looser band of half-width 0.02, because the $[-0.01, 0.01]$ band is a declared convention, not a measured quantity. Only null endpoints change, since a directional contract tests its interval against zero. Table 8 gives the contract failures of the confirmation matrix and of both repair experiments.

The separation between architectures persists across all bands. DKT fails four null endpoints at 0.005 and none at the declared or wider bands, so the perfect DKT score depends as much on the declared band as on the model. SAKT fails 14 contracts at 0.005, eight at the declared band, and seven at 0.02. The seven failures at 0.02 are the three directional $A \to B$ targets and the four recency endpoints. The band does not affect the directional targets, and the recency intervals extend beyond ± 0.02. The promotion decision also persists. In the frozen pilot, PCR leaves two failures versus seven for the baseline at 0.005 and one versus four at both wider bands. In the multi-seed repair confirmation, PCR leaves fewer failures than its baseline under every logging policy at every band, and under blocked logging at 0.02 it passes all eight contracts. The promotion gate therefore admits PCR at every examined band. The gate refuses the misspecified arm of Appendix G.3 at both tighter bands but admits that arm at 0.02.

Table 8: Contract failures across three equivalence bands in the confirmation matrix (24 endpoints per model), frozen pilot (16 endpoints), and multi-seed repair confirmation (eight endpoints per logging policy)

		Band		
Experiment	Model or arm	0.005	0.01	0.02
Confirmation matrix	DKT	4	0	0
	SAKT	14	8	7
Frozen pilot	Baseline	7	4	4
	PCR	2	1	1
Multi-seed, uniform	Baseline	4	4	4
	PCR	2	2	1
Multi-seed, blocked	Baseline	6	4	4
	PCR	3	2	0
Multi-seed, adaptive	Baseline	4	3	2
	PCR	3	2	1

G PCR repair experiments

Table 9 reports all 16 endpoint contracts of the exploratory uniform-policy repair pilot with one dataset seed and one model seed. Each endpoint cell shows baseline $\to$ PCR, and a superscript F marks an arm that fails its endpoint contract. AUC drop is baseline AUC minus PCR AUC, so a negative drop means PCR raises AUC. All entries are point estimates rather than population confidence intervals.

Table 9: Complete frozen repair pilot on offset-8 audit identities

Model	Regime	Source: base→PCR	Target: base→PCR	AUC: base→PCR	Drop
DKT	$A \to B$	+0.0024→+0.0016	+0.0128→+0.0092	0.881→0.880	+0.00048
DKT	$B \to A$	-0.0189→-0.0293	-0.0045→-0.0039	0.877→0.876	+0.00037
DKT	Independent	+0.0030→+0.0005	-0.0010→+0.0015	0.854→0.853	+0.00107
DKT	Recency	-0.0050→-0.0049	+0.0015→+0.0015	0.882→0.882	+0.00006
SAKT	$A \to B$	+0.0070→-0.0013	$-0.0679^{\mathbf{F}}$→+0.0462	0.874→0.873	+0.00095
SAKT	$B \to A$	$+0.0006^{\mathbf{F}}$→-0.0147	-0.0021→-0.0007	0.873→0.873	-0.00008
SAKT	Independent	+0.0047→+0.0037	-0.0082→-0.0018	0.849→0.848	+0.00130
SAKT	Recency	$+0.0374^{\mathbf{F}}$→$+0.0371^{\mathbf{F}}$	$-0.0254^{\mathbf{F}}$→-0.0068	0.648→0.643	+0.00491

The baseline fails four endpoint contracts, whereas PCR fails one, the SAKT recency-source null contract. The largest AUC drop, 0.00491, is below the fixed 0.01 guard. Because the table shows every cell, the reported 15/16 result cannot hide an AUC-poor or contract-poor subgroup.

G.1 Identity separation and objective inputs

The identity separation applies only to the PCR auxiliary loss, not to factual training. Factual BCE uses the complete logged training batch, including item identities that may also occur in ordinary test prediction. PCR constructs matched auxiliary trajectories with item offset 0, and only these trajectories exclude offset-8 audit identities. Audit identities can therefore occur in model training. The leakage claim is only that they are absent from the PCR auxiliary loss, which directly encodes the evaluated contracts. Table 10 lists the configured role of each identity.

Table 10: Configured identity roles in both repair experiments. Each concept has 16 items. Values are 1-based item indices within a concept, shifted by 16 for target-concept items.

Role	Early block	Delayed suffix	Probes
PCR auxiliary loss (offset 0)	1–2	3–4	5–6
Frozen audit (offset 8)	9–10	11–12	13–14

G.2 Multi-seed repair confirmation

Because the pilot fixes one dataset seed, one model seed, and uniform logging, a separate repair confirmation varies all three. It trains SAKT alone under each of the three logging policies on five fresh dataset seeds and three fresh model seeds. The matched baseline and PCR arms use the pilot objective, optimization budget, sequence length, and audit. As in the confirmation matrix, the analysis averages model seeds within each dataset before bootstrapping the five dataset means. Each policy has its own aggregate and its own promotion decision. Table 11 reports the eight endpoint contracts of each policy, each scored against the contract of its regime.

PCR passes the same two-part promotion gate under every policy. Failed contracts fall from 4/8 to 2/8 under uniform and blocked logging and from 3/8 to 2/8 under adaptive logging. The worst matched test-AUC drops under uniform, blocked, and adaptive logging are 0.00502, 0.00525, and 0.00013, all below the 0.01 predictive guard. PCR passes every contract that its matched baseline passes.

Under uniform logging, PCR repairs both failed transfer contracts. The forward-transfer target, whose baseline interval spans zero, moves to $+0.0119$ with an interval above zero. The reverse-transfer source moves from a wrong-signed $+0.0112$ to -0.0106 with an interval below zero. Both recency endpoint effects shrink, but their intervals still extend outside the band. The source moves from -0.0093 to $+0.0023$ with an interval reaching $+0.0110$, and the target moves from $+0.0381$ to $+0.0164$.

Blocked logging repeats this pattern. PCR moves the forward-transfer target from $+0.0021$ to $+0.0313$ and the reverse-transfer source from -0.0005 to -0.0261. Both baseline intervals span zero, whereas both PCR intervals lie on the prescribed side of zero. The recency target shrinks from $+0.0368$ to $+0.0027$, but its interval reaches $+0.0109$. The recency source moves from -0.0133 to $+0.0092$, and its interval still extends beyond the band.

Under adaptive logging, the baseline already satisfies the forward-transfer contract. PCR repairs the recency source, whose baseline interval reaches $+0.0103$. The reverse-transfer source changes from a wrong-signed $+0.0102$ to -0.0017, but its interval still spans zero. The recency target shrinks from $+0.0126$ to $+0.0067$, and its interval reaches $+0.0137$.

PCR repairs three of four failures in the pilot and five of 11 across the three confirmation policies. At each of the 11 failed endpoints, the mean effect moves toward its contract, and each of the 13 passing endpoints still passes. The multi-seed confirmation thus fixes the extent of repair at five of 11, while the direction of repair and the predictive guard persist across seeds and logging policies.

Table 11: Multi-seed repair confirmation on SAKT under three logging policies (five dataset seeds, three model seeds). Superscript F marks a failed contract.

Regime	Endpoint	Baseline effect [95% CI]	PCR effect [95% CI]
Uniform logging			
$A \to B$	Source	$+0.0020\,[+0.0004, +0.0042]$	$+0.0029\,[+0.0015, +0.0050]$
$A \to B$	Target	$-0.0058\,[-0.0231, +0.0104]^{\mathbf{F}}$	$+0.0119\,[+0.0054, +0.0182]$
$B \to A$	Source	$+0.0112\,[+0.0045, +0.0187]^{\mathbf{F}}$	$-0.0106\,[-0.0192, -0.0020]$
$B \to A$	Target	$-0.0004\,[-0.0013, +0.0006]$	$+0.0018\,[+0.0011, +0.0025]$
Independent	Source	$+0.0017\,[+0.0004, +0.0032]$	$+0.0009\,[-0.0002, +0.0025]$
Independent	Target	$+0.0008\,[+0.0003, +0.0015]$	$+0.0005\,[-0.0006, +0.0017]$
Recency	Source	$-0.0093\,[-0.0330, +0.0100]^{\mathbf{F}}$	$+0.0023\,[-0.0081, +0.0110]^{\mathbf{F}}$
Recency	Target	$+0.0381\,[+0.0104, +0.0657]^{\mathbf{F}}$	$+0.0164\,[+0.0013, +0.0307]^{\mathbf{F}}$
Blocked logging			
$A \to B$	Source	$+0.0007\,[-0.0013, +0.0027]$	$+0.0008\,[-0.0010, +0.0022]$
$A \to B$	Target	$+0.0021\,[-0.0143, +0.0185]^{\mathbf{F}}$	$+0.0313\,[+0.0210, +0.0427]$
$B \to A$	Source	$-0.0005\,[-0.0168, +0.0159]^{\mathbf{F}}$	$-0.0261\,[-0.0349, -0.0173]$
$B \to A$	Target	$+0.0003\,[-0.0014, +0.0021]$	$+0.0028\,[+0.0019, +0.0040]$
Independent	Source	$-0.0029\,[-0.0054, +0.0003]$	$-0.0016\,[-0.0030, -0.0003]$
Independent	Target	$+0.0052\,[+0.0045, +0.0058]$	$+0.0038\,[+0.0020, +0.0055]$
Recency	Source	$-0.0133\,[-0.0249, -0.0034]^{\mathbf{F}}$	$+0.0092\,[+0.0023, +0.0155]^{\mathbf{F}}$
Recency	Target	$+0.0368\,[+0.0210, +0.0526]^{\mathbf{F}}$	$+0.0027\,[-0.0055, +0.0109]^{\mathbf{F}}$
Adaptive logging			
$A \to B$	Source	$-0.0030\,[-0.0056, +0.0000]$	$-0.0008\,[-0.0013, -0.0002]$
$A \to B$	Target	$+0.0062\,[+0.0038, +0.0087]$	$+0.0099\,[+0.0081, +0.0117]$
$B \to A$	Source	$+0.0102\,[+0.0093, +0.0110]^{\mathbf{F}}$	$-0.0017\,[-0.0064, +0.0032]^{\mathbf{F}}$
$B \to A$	Target	$-0.0013\,[-0.0028, +0.0007]$	$+0.0000\,[-0.0017, +0.0029]$
Independent	Source	$-0.0020\,[-0.0031, -0.0004]$	$-0.0014\,[-0.0029, +0.0001]$
Independent	Target	$+0.0015\,[-0.0006, +0.0037]$	$+0.0000\,[-0.0015, +0.0015]$
Recency	Source	$+0.0022\,[-0.0071, +0.0103]^{\mathbf{F}}$	$-0.0005\,[-0.0097, +0.0090]$
Recency	Target	$+0.0126\,[+0.0016, +0.0232]^{\mathbf{F}}$	$+0.0067\,[-0.0005, +0.0137]^{\mathbf{F}}$

G.3 Misspecified-contract ablation

Because PCR teaches whatever contract it receives, the ablation supplies misspecified contracts. On the same seeds under uniform logging, a third arm trains SAKT with these misspecified contracts: exchanged transfer contracts, a directional prior on the independent regime, and a reverse prior on the recency regime. The audit still scores the true contract of each regime.

The promotion gate refuses the misspecified arm. Failed contracts stay at 4/8, so the misspecified contract repairs nothing. The worst drop in matched test AUC rises to 0.00696, compared with 0.00502 under the specified contract. The two endpoints that the specified contract repairs still fail under the misspecified contract. The forward-transfer target reaches -0.0015, whereas it reaches $+0.0119$ under the specified contract. The reverse-transfer source stays positive at $+0.0071$, whereas its endpoint contract requires an interval below zero. The refusal depends on the null band. At 0.005 the arm fails 6/8 against 4/8 for the baseline, whereas at 0.02 it fails 3/8 against 4/8 and is promoted.

A process improvement is therefore a property of the contract, not of the loss. Under a misspecified contract, the same loss repairs nothing at the declared band and reduces predictive fidelity more. A

predictive check alone does not reject such a model, because its AUC drop stays inside the guard. The audit cannot certify that the supplied contract is correct and can only show that a model fails this contract. The promotion gate requires strictly fewer failures and therefore refuses the model at the declared band.

H TRAINING, SEEDS, AND REPRODUCIBILITY DETAILS

All learned models on synthetic data use length-64 sequences and the same training configuration: a 70/10/20 train/validation/test split (split seed 42), learning rate 10^{-3}, batch size 64, at most 20 epochs, early-stopping patience 5, gradient-norm clipping at 5, embedding dimension 64, dropout 0.2, and checkpoint selection by validation AUC. Table 12 lists architecture-specific settings copied from the training receipts.

Table 12: Architecture-specific synthetic training settings

Model	Configuration
DKT	recurrent hidden size 200
SAKT	model dimension 64; 4 attention heads

PCR uses $\lambda = 1$, directional margin $m = 0.01$, null tolerance $\epsilon = 0.01$, prefix length 20, block size 2, probe size 2, and a 16-learner sub-batch. The five-dataset aggregates use 2,000 bootstrap resamples, and the one-dataset repair receipt uses 10,000. Because a one-dataset interval collapses to the observed point, the repair pilot table is exploratory. The multi-seed repair confirmation in Appendix G.2 carries the promotion claim that survives fresh seeds. Table 13 separates development, confirmation, and repair evidence.

Table 13: Separation of development, confirmation, and repair evidence

Stage	Data seeds	Model seeds	Policies	Checkpoints	Use
Excluded pilot	0	42	Uniform	8	debugging and hypothesis formation
Confirmation	101–105	201–205	3	600	architecture/policy contract test
Repair pilot	301	401	Uniform	16	exploratory baseline/PCR comparison
Repair confirmation	312–316	412–414	3	420	multi-seed SAKT arms; misspecified arm under uniform logging

The 600 confirmation checkpoints span four regimes, three policies, five generated datasets, five model seeds, and two published backbones. The 16 repair-pilot checkpoints span four regimes, two backbones, and two training arms. The 420 repair-confirmation checkpoints span four regimes, five dataset seeds, three model seeds, and one backbone. Baseline and PCR arms run under each of the three policies, and the misspecified-contract arm runs under uniform logging. Development, confirmation, and repair estimates are never pooled. Dataset SHA-256 digests bind each run to its generated interactions, and run receipts also bind training specifications, checkpoints, metric files, evaluation indexes, and aggregate JSON files. The aggregator recomputes contracts from endpoint rows and checks that the expected regime–policy–seed–model grid is complete.

H.1 REPRODUCTION ORDER

A clean reproduction proceeds in this order:

1. Generate all synthetic datasets using the recorded regime, logging policy, and dataset seed, then verify the dataset digests.
2. Train the baseline checkpoint grid from immutable training specifications, then verify that every expected run succeeds.
3. Evaluate the exact state-machine oracle using the same matched operator and stop if any oracle endpoint fails.
4. Evaluate the learned checkpoints, aggregate model seeds within each dataset, and bootstrap the dataset-seed means.

5. Train the matched baseline and PCR repair arms with dataset seed 301 and model seed 401, evaluate both arms on offset-8 identities, and recompute the two-part promotion decision.
6. Repeat the baseline and PCR arms under all three logging policies, and the misspecified arm under uniform logging, with dataset seeds 312–316 and model seeds 412–414.
7. Compile tables and figures from aggregate receipts rather than console logs.

I Extended related work and conceptual positioning

Knowledge tracing and predictive evaluation. Knowledge tracing spans explicit latent-mastery models and recurrent, memory, attention, graph, and hybrid models, as reviewed by Abdelrahman et al. (2023). Large benchmark efforts improve preprocessing and comparison discipline. The pyKT library standardizes datasets, models, and evaluation scenarios (Liu et al., 2022), and XES3G5M adds rich question and concept metadata at scale (Liu et al., 2023b). Empirical comparisons show that architecture, hyperparameters, metrics, seeds, and dataset scale can all change predictive rankings (Sarsa et al., 2022; Kim & Kim, 2026). These studies motivate careful predictive benchmarking. The matched audit instead tests whether the state transition of a predictor under a controlled curriculum permutation satisfies a prespecified process contract. The contribution pairs directional and equivalence semantics with matched transformations, learner-level uncertainty, and a predictive promotion guard.

Invariance across interventions and environments. Invariant causal prediction identifies causal predictors from stability across environments and interventions under explicit assumptions (Gu et al., 2025). Invariant risk minimization seeks representations whose optimal classifier is stable across training environments (Norman & Meir, 2025). The matched audit evaluates a model under structured environment changes but does not infer a causal graph or claim that relations in logged educational data are causal. Instead, a process contract specifies the transformation and expected endpoint response. Unlike an invariance objective, the audit prespecifies directional and practically null endpoint responses and decides each one after training. On real logs, a failure provides evidence about model semantics; only the transparent simulator supplies a known causal data-generating mechanism.

Decision-aware simulator learning. Long-standing work on model-based decision making recognizes that generic predictive loss need not align with downstream planning. Value-aware model learning changes the model objective to reflect the decision problem (Voelcker et al., 2025). Objective-mismatch studies show that one-step likelihood may be weakly related to control performance (Wei et al., 2024). Multi-step analyses characterize how local model error can compound during rollout (Lambert et al., 2022). World-model training can close the gap between next-state prediction and use in planning through train-time data synthesis (Parthasarathy et al., 2025). PCR complements prior approaches without optimizing a task value function. It retains a predictive guard and adds qualitative, intervention-level transition constraints that make the intended semantics inspectable before a learner model is used for planning.

J Claim boundaries and additional stress tests

The present evidence supports three bounded claims. First, logged next-response accuracy alone does not identify matched order effects. On ASSISTments 2009, each of five architectures yields positive and negative directional-minus-neutral contrasts across three training seeds while its test AUC varies by at most 0.0063. Second, under the transparent simulator and the frozen protocol, contract failures depend on architecture and logging policy even at high AUC. Third, PCR reduces failures on item identities outside the auxiliary loss while satisfying the predictive guard. Failures fall from 4/16 to 1/16 in the exploratory repair pilot and from 11/24 to 6/24 for SAKT across fresh seeds and three logging policies. The promotion gate refuses PCR trained against a misspecified contract. The evidence does not show that the simulator contracts are universal laws of learning, that the real-log effects are causal effects on students, or that the audit can certify a contract as correct.

The evaluation challenges the method with a multi-seed repair confirmation under three logging policies, a misspecified contract, and alternative equivalence bands. The evidence therefore supports a feasibility claim under the evaluated contracts. A deployment claim needs evidence on noisy and partially correct prerequisite graphs and on longer horizons with forgetting and hint events. It also needs prospective educational interventions that separate human causal effects from process validity.