Extended Abstract

Motivation. Reinforcement learning formulations of tutoring require a reward function, and prior work focused on academic subjects with inherent reward structure: predicted performance on a terminal assessment. Personal finance has no such standardized assessment, and its content keeps shifting as tax rules, products, and markets change. The optimization target becomes an open design decision. This project asks whether that decision changes how a learned tutoring policy teaches, not only how well it scores.

Method. We build a controlled, offline RL testbed in which the reward is the single varying factor. A tutor selects (concept, difficulty) practice items over 50-step sessions on a 16-concept personal-finance curriculum with an explicit prerequisite graph. From a fixed pool of 10,002 logged sessions per student simulator, we relabel terminal rewards under three designs. The first is expected accuracy on held-out quiz probes, the second is expected success on multi-concept scenarios scored conjunctively, and the third is a Bradley–Terry reward model learned from pairwise session comparisons by an LLM judge under a short teaching rubric, with every comparison judged in both presentation orders to cancel position bias. Policies are evaluated on a ladder of three student simulators that vary cross-concept structure, from independent Bayesian knowledge tracing (BKT), to Coupled-BKT with hand-coded prerequisite transfer, to an LSTM knowledge tracer (DKT) trained on Coupled-BKT sessions.

Implementation. Offline datasets are generated by three scripted behavior policies, and training uses a discrete-action version of implicit Q-learning, implemented from scratch, with behavior cloning and an AWR variant as reference points. Each simulator receives its own preference reward model, a small MLP over a 52-feature session summary with 89–97% held-out agreement with the judge. The grid spans three rewards, three simulators, and ten seeds, giving 90 primary policies plus 150 reference runs.

Results. We find three things. First, the reward reshapes teaching behavior, and the direction depends on the learner model. Relative to quiz-trained policies, preference-trained policies teach more broadly on BKT (concept entropy +0.93 nats, $d = 2.4$) and Coupled-BKT (+0.45, $d = 1.2$) but more narrowly on DKT (-0.59, $d = -4.3$). All three contrasts survive Holm correction, while pooling across simulators masks the effect entirely (pooled $p = 0.054$). Second, the learned preference reward transfers to the objectives it never saw. Preference-trained policies beat quiz-trained policies on quiz accuracy (0.458 vs. 0.394, $d = 1.2$) and scenario-trained policies on scenario success (0.194 vs. 0.143, $d = 1.4$), exactly on the two simulators (BKT and Coupled-BKT) where the preference reward also broadens the curriculum. Third, the judge itself prefers the preference policy's sessions in 80–100% of order-robust head-to-head comparisons on every simulator, including DKT, where that policy teaches narrowly, so the trained policies satisfied the judge rather than exploiting blind spots of the learned reward model.

Discussion. Behavior cloning on identical data shows neither effect, so the differences trace to the reward signal rather than the data. A pooled analysis would have called the central behavioral effect marginal, so simulator-pooled claims can hide real, opposite-signed effects. All learners are simulated, and the judge's standard is untested against human learning outcomes.

Conclusion. We recommend treating reward design as an explicit, audited decision in tutoring RL, evaluating policies behaviorally and not only by attained score, and always disaggregating behavioral claims by learner model. Promising next steps are human learners, sequence-level reward models, a wider range of judges and rubrics, and a mechanistic account of the DKT reversal.

Beyond Test Scores: Reward Design for Offline RL in Personal Finance Tutoring

Stanford CS224R Project

Daniel Argento
Department of Computer Science
Stanford University
dargento@stanford.edu

Stella Wu
Departments of Computer Science & Philosophy
Stanford University
stellaw@stanford.edu

Abstract

Reinforcement learning tutoring systems inherit their reward from a domain's terminal assessment—a proxy unavailable in personal finance, where no standardized exam exists. We study whether this open design decision changes *how* a policy teaches, not only how well it scores. In a controlled offline setting, we relabel a fixed pool of logged sessions under three reward designs—quiz accuracy, conjunctive scenario success, and a Bradley–Terry preference reward from an LLM judge—and train discrete IQL policies on a 16-concept curriculum evaluated against a ladder of three student simulators (BKT, Coupled-BKT, DKT). Reward choice reshapes curriculum breadth, but the direction depends on the learner model: preference-trained policies broaden teaching on the Bayesian simulators (concept entropy +0.93 nats, $d = 2.4$) and narrow it on DKT (-0.59 nats, $d = -4.3$); pooling across simulators masks both effects entirely. The preference reward also transfers to held-out metrics where it broadens the curriculum, and the LLM judge endorses the resulting sessions in 80–100% of head-to-head comparisons. Behavior cloning on the same data shows none of these effects, localizing them to the reward signal. Reward design is a behavioral lever that should be explicitly audited, and simulator-pooled claims should always be disaggregated by learner model.

1 Introduction

Intelligent tutoring systems deliver individualized instruction at scale [17]. Existing systems are designed for domains with a well-defined, sequence-based curriculum and concrete end-goals, such as mastery of a fixed syllabus or performance on a terminal exam [6], and reinforcement learning formulations of tutoring inherit this dependence, since the reward is typically derived from the curriculum's terminal assessment. Personal finance education does not share these properties. The domain has no standardized assessment playing the role a standardized exam plays in K–12 mathematics, and its content changes over time with new laws, financial products, and market conditions, so the optimization target, elsewhere largely determined by the curriculum itself, becomes an open design question. What should an RL tutor optimize for?

There are several hypotheses: the student's test performance, the student's ability to apply knowledge across interrelated concepts, or a holistic assessment of teaching quality, the approach taken by preference-based reward learning [4]. It is unclear whether policies trained on these objectives teach similarly, and the distinction matters because an objective might change not only the quality of instruction but which concepts a student is taught at all.

Existing work does not directly address this question. Knowledge tracing estimates a learner's concept mastery from response sequences [5, 14] but does not specify what objective a teaching policy should pursue. Work on RL for instructional sequencing identifies reward specification as a central open difficulty [6] but does not empirically compare reward designs, and prior tutoring work in personal finance inherits its objectives from the frameworks it builds on (Section 2).

We study this question in a controlled offline setting where the reward is the only thing that varies. A tutoring policy selects (concept, difficulty) practice items over 50-step sessions from a 16-concept personal-finance curriculum with an explicit prerequisite graph, mirroring deployed systems, which train from logged data instead of online experiments on real students. From a fixed pool of 10,002 pre-collected sessions per simulator, we relabel only the terminal reward under three designs: expected accuracy on held-out quiz probes, expected success on multi-concept scenarios scored conjunctively, and a Bradley–Terry reward model trained on pairwise session comparisons produced by an LLM judge [4, 1] with position-bias control. Policies are trained with a discrete adaptation of implicit Q-learning [9] and evaluated on a ladder of three student simulators that vary the presence and provenance of cross-concept structure (Section 3.2), using curriculum-structure metrics covering breadth, prerequisite ordering, and difficulty progression. Every configuration runs with ten seeds, and all behavioral claims are tested both pooled across simulators and per simulator.

Swapping the reward changes teaching behavior, and the direction of the change depends on the learner model. Behavior-cloning policies trained on the same data show none of these effects. This paper contributes:

1. a controlled offline testbed in which the reward is the single varying factor, spanning a 16-concept curriculum and a ladder of three student simulators;
2. a position-bias-controlled RLAIF pipeline for teaching preferences, with a separate Bradley–Terry reward model per simulator (judge agreement 89–97%);
3. evidence that reward choice reshapes curriculum structure with a direction that depends on the learner model, that pooling across simulators can mask the effect entirely, and that the preference reward transfers to the hand-specified metrics where it broadens the curriculum, with the judge preferring the resulting policies on every simulator.

2 Related Work

Knowledge tracing and student simulation. Knowledge tracing infers a learner's latent mastery from observed responses. Bayesian knowledge tracing (BKT) models each concept as an independent two-state hidden Markov model [5], and deep knowledge tracing (DKT) replaces this with a recurrent network that learns cross-concept dependencies directly from response data [14]. Where this literature builds mastery estimators, we use these models as student simulators that generate training data and serve as evaluation environments. From psychometrics we adopt conjunctive scoring [8], where the student must get every required skill right at once, as the form of our multi-concept scenario reward.

Reinforcement learning for instructional sequencing. A substantial line of work learns or plans teaching policies, including teaching as POMDP planning [15], offline policy evaluation for educational games [11], and deep RL for scheduling learning activities [2]. Doroudi et al. [6] survey this literature and identify reward specification as a central open difficulty, observing that most systems optimize a default objective such as predicted post-test performance. To our knowledge, no prior work answers that question empirically by comparing reward designs under a fixed dataset, algorithm, and evaluation pipeline, the comparison we run here.

Tutoring systems for personal finance. Personal finance remains sparsely studied within the intelligent tutoring literature. Tan [16] conducts a formative study toward a personal-finance tutoring system, eliciting learners' metacognitive approaches through surveys and think-aloud sessions

with a low-fidelity prototype, and Johnson and Solberg [7] design an adaptive instructional system for financial literacy that personalizes content by career stage and life events instead of tracking mastery. Both inherit their instructional objectives from the frameworks they build on rather than examining the objective itself, and neither learns a teaching policy with reinforcement learning.

Offline reinforcement learning. We use implicit Q-learning (IQL) [9], which avoids the offline setting's central failure mode of value overestimation on out-of-distribution actions [10] by fitting an upper expectile of the value function over dataset actions, with policies extracted by advantage-weighted regression [13]. Our contribution is orthogonal to algorithm design.

Learning rewards from preferences. Rather than hand-specifying a reward, preference-based methods fit a Bradley–Terry model [3] to pairwise trajectory comparisons [4]. Reinforcement learning from AI feedback replaces human comparison labels with judgments produced by a language model operating under a fixed rubric [1]. Preference-derived rewards substantially shape the behavior of large language models [12], but whether they have analogous effects on sequential decision-making policies in education has not been examined. Unlike RLHF for language models, where behavior is a text distribution, tutoring behavior is a curriculum with directly measurable pedagogical structure.

3 Methods

3.1 Problem Formulation

We model a tutoring session as a finite-horizon Markov decision process. Let $\mathcal{C}$ denote a set of $K = 16$ personal-finance concepts connected by a prerequisite graph G, a directed acyclic graph (DAG) with 21 edges shown in Figure A1 (appendix), and let $\mathcal{D} = \{1, 2, 3\}$ denote item difficulty levels (easy, medium, hard). The concepts were selected by surveying topics that appear most prominently across introductory personal-finance resources, ranging from foundational skills such as *budgeting* to capstone topics such as *retirement planning*. The prerequisite graph encodes logical dependencies between topics, for example that *compound interest* builds on *simple interest*. Specifying such a graph is compatible with our claim that the domain lacks a canonical curriculum. What is missing is a standard terminal exam to inherit a reward from, and the graph is a design choice for the simulators, not a prescription for how finance must be taught.

Action space. At each step the tutor selects a pair $(c, d) \in \mathcal{C} \times \mathcal{D}$, giving 48 discrete actions. A held-out set of 48 quiz probes (one per action cell) is reserved for evaluation and never used during practice.

State space. The state $s_t \in [0, 1]^K$ is the student simulator's per-concept knowledge estimate after t practice actions. For the Bayesian simulators this is the posterior mastery probability, the simulator's complete internal state, so the problem is fully observed. For DKT the policy sees only the network's 16 per-concept predictions rather than its full internal LSTM state, making observability approximate.

Dynamics and horizon. Upon action (c_t, d_t), the learner's response is sampled as $y_t \sim \text{Bernoulli}(p_t)$ with $p_t = P(\text{correct} \mid s_t, c_t, d_t)$ given by the simulator, which then updates its state given (c_t, y_t, d_t). Episodes run for a fixed horizon of $T = 50$ steps with no early termination. The reward is issued only at episode end (Section 3.3), and intermediate rewards are zero.

3.2 Student Simulators

Conclusions about reward design should not depend on the choice of learner model, so we evaluate against a ladder of three simulators that vary the presence and provenance of cross-concept structure, from none (independent BKT) to hand-coded (Coupled-BKT) to learned from data (DKT). All three plug into the environment the same way, so they are interchangeable.

Bayesian knowledge tracing (BKT). Classical BKT [5] models each concept c as an independent two-state hidden Markov model. The learner either has mastered the concept or has not, and the simulator tracks m_c, its belief that the learner is in the mastered state. A mastered concept can still be answered incorrectly with slip probability p_c^{slip}, and an unmastered one correctly with guess probability p_c^{guess}, so the probability of a correct response at difficulty d is

$$P(\text{correct} \mid m_c, d) \;=\; m_c\,\big(1-\sigma_c(d)\big) \;+\; (1-m_c)\,p_c^{\text{guess}}, \tag{1}$$

where $\sigma_c(d)$ is the difficulty-adjusted slip probability defined below. After observing response y, mastery is updated by the Bayesian posterior followed by a learning transition

$$m_c \;\leftarrow\; \tilde{m}_c + \big(1-\tilde{m}_c\big)\,\ell_c(d), \tag{2}$$

where $\tilde{m}_c = P(\text{mastered} \mid y)$ is the posterior and $\ell_c(d)$ the difficulty-adjusted learning rate. Parameters are set per concept to reflect the topics themselves. *Budgeting* is quick to pick up ($p^{\text{learn}} = 0.20$, $p^{\text{slip}} = 0.08$), while *retirement planning* is slow and error-prone ($p^{\text{learn}} = 0.08$, $p^{\text{slip}} = 0.18$), and the guess probability is 0.2 throughout. BKT is the transparent base case. Equations 1 and 2 fully specify the learner, so any behavioral effect observed under BKT must come from the rewards and the policy. Its limitation is that practicing a concept never affects any other concept, so the prerequisite graph cannot influence learning at all.

Coupled-BKT. To give the prerequisite graph causal force, Coupled-BKT augments BKT with a one-hop transfer rule. When the learner answers an item on concept c correctly, every direct dependent c' (every c' with $c \in \text{prereq}(c')$) receives a fractional learning increment

$$m_{c'} \;\leftarrow\; m_{c'} + \lambda\, p_{c'}^{\text{learn}}\big(1-m_{c'}\big), \qquad \lambda = 0.5. \tag{3}$$

Transfer is asymmetric (incorrect responses trigger no transfer), one-hop (no transitive cascade), and unmodulated by difficulty. A single global strength λ applies to every edge, scaled only by the dependent's own learning rate, and the increment equals half the learning-transition gain of a single correct easy practice on the dependent itself. Because the transfer rule is hand-coded, it serves as exactly specified ground truth for cross-concept structure, and we use it as the data-generating model for training the learned simulator below. The downside is that this rule may not reflect how learning actually transfers between topics, so Coupled-BKT should not be the sole evaluator of policies whose behavior depends on it.

Deep knowledge tracing (DKT). The top rung replaces specified structure with learned structure, because deployed learner models are fit from data, not designed. Following Piech et al. [14], a single-layer LSTM with 128 hidden units consumes a one-hot encoding of each (concept, response) pair and outputs per-concept correctness probabilities through a sigmoid head. The network is trained on a dedicated pool of 5,001 Coupled-BKT sessions (Section 3.4) using the standard next-step prediction objective. Training uses Adam (learning rate 10^{-3}, 30 epochs, batch size 64) with checkpoint selection by held-out AUC on 10% of sessions (best validation AUC 0.680). Doubling

the pool improves AUC by only 0.005, so the model operates near the noise ceiling of the guess-and-slip response process. At simulation time the network is frozen and its hidden state plays the role of the learner's knowledge state, advanced by each observation. Whether the LSTM actually internalizes the prerequisite structure of its training data is an empirical question, which we test with a behavioral probe (Section 5). The cost of this rung is opacity. There is no interpretable mastery quantity, so we use the predicted correctness vector as the belief state, and the dynamics approximate the true generative process rather than reproduce it.

Difficulty modulation. All three simulators make harder items both riskier and less instructive, sharing a single slip-boost constant $\kappa = 0.05$. For the Bayesian simulators, difficulty enters in two places. The slip probability rises as $\sigma_c(d) = \min\big(1,\, p_c^{\text{slip}} + \kappa\,(d-1)\big)$, and the learning rate decays as $\ell_c(d) = p_c^{\text{learn}} \cdot 0.85^{\,(d-1)}$, so a hard item slips roughly ten percentage points more and teaches roughly 28% less per exposure than an easy one. For DKT, difficulty is applied after the network output as $P(\text{correct} \mid c, d) = \text{clip}\big(\hat{p}_c - \kappa\,(d-1),\, 0,\, 1\big)$, and the LSTM itself never observes difficulty. Holding the difficulty mechanism fixed and external to the learned model keeps cross-concept structure as the single factor that varies across the ladder. The residual approximation shows BKT slip boost shrinks with low mastery while DKT subtracts the full $\kappa\,(d-1)$, slightly harsher on weak students.

3.3 Reward Designs

Each terminal reward is a function of the completed session $\xi = \big(s_t, c_t, d_t, y_t\big)_{t=1}^{T}$ and of the simulator's final state s_T, which reward functions may probe without advancing it.

Quiz accuracy. The first objective follows tutoring RL convention and rewards measured mastery. At episode end, the learner is probed on the 48 held-out quiz cells, and the reward is the mean expected accuracy

$$R_{\text{quiz}}(\xi) \;=\; \frac{1}{|Q|} \sum_{q \in Q} P\big(\text{correct} \mid s_T, c_q, d_q\big), \tag{4}$$

using the simulator's response probability rather than sampled outcomes, which removes response noise from the training signal. This objective values per-item recall and is indifferent to how mastery is distributed across concepts beyond its effect on the average.

Scenario success. The second objective rewards integrated application of multiple concepts, the way real financial decisions tend to draw on several topics at once. We hand-specify 40 scenarios, each consisting of a required concept set S_j with two or three members and its own difficulty d_j, applied to every concept in the set. Combinations pair topics that co-occur in realistic decisions, for example *budgeting* with *emergency fund*. Success is modeled conjunctively [8], meaning the learner must get every required concept right, so

$$R_{\text{scen}}(\xi) \;=\; \frac{1}{40} \sum_{j=1}^{40} \prod_{c \,\in\, S_j} P\big(\text{correct} \mid s_T, c, d_j\big). \tag{5}$$

The product makes each scenario's score limited by its weakest required concept, so the objective pressures the policy to raise weak links rather than maximize average mastery.

Learned preference reward. The third objective is learned from pairwise comparisons instead of specified by hand, following the preference-learning paradigm [4] with an LLM judge in place of

human annotators [1]. For each simulator we sample 500 session pairs from its offline dataset (70% across behavior policies, 30% within one policy), render each session as a structured text summary, and ask the judge (`claude-haiku-4-5`, temperature 0) which session shows better teaching. The rubric, reproduced in full in the appendix, is a short, unordered statement of three aspects of good teaching: mastery across several concepts, prerequisite ordering, and challenge matched to the student. To control for position bias, every pair is judged twice with presentation order swapped, and pairs where the preference flips are excluded from training (order-robust verdicts on 76%, 75%, and 87% of pairs for BKT, Coupled-BKT, and DKT respectively). A separate reward model r_ϕ is then fit per simulator under the Bradley–Terry likelihood [3], which learns a scalar score that is higher for the sessions the judge tends to prefer,

$$\mathcal{L}(\phi) \;=\; -\,\mathbb{E}_{(\xi^+,\,\xi^-)}\Big[\log\sigma\big(r_\phi(\xi^+) - r_\phi(\xi^-)\big)\Big], \tag{6}$$

where ξ^+ is the judge-preferred session. Each model is a two-hidden-layer MLP (128 units) whose input is a fixed 52-dimensional session summary: the final mastery vector (16 values), the fraction of practice steps spent on each concept (16), the per-concept correctness rates (16), and four scalars (overall correctness rate, revisit rate, mean difficulty, episode length). We use a fixed summary instead of a sequence model to keep the reward simple to fit and audit. Held-out agreement with the judge is 97%, 89%, and 93% for the BKT, Coupled-BKT, and DKT models, and hand-audited samples of judged pairs against archived raw responses confirm the judge followed the rubric. Finally, each simulator's dataset is relabeled by its own reward model, so the preference reward is always computed by a model matched to the learner being taught.

3.4 Offline Policy Learning

Dataset construction. For each simulator we collect a fixed offline dataset of 10,002 sessions (seed 42), generated in equal proportion by three scripted behavior policies. The first selects uniformly at random over the 48 actions, the second practices only concepts whose prerequisites exceed mastery 0.5 at uniform difficulty, and the third advances difficulty from easy to medium to hard over thirds of the episode with uniform concepts. The mixture provides both coverage and structured, curriculum-like trajectories, and is disjoint from the 5,001-session pool (seed 100) used to train the DKT simulator. Each dataset is flattened to $10{,}002 \times 50 = 500{,}100$ transitions with the relabeled reward placed at the terminal transition. Swapping which of the three rewards labels the dataset is the experiment's only manipulation. States, actions, and learner responses are identical across conditions.

Discrete IQL. We implement implicit Q-learning [9] from scratch, adapted to the discrete action space. $Q(s,\cdot)$ outputs all 48 action values, $V(s)$ is scalar, and $\pi(\cdot \mid s)$ is categorical (each a two-hidden-layer MLP with 128 units). IQL avoids querying out-of-distribution actions by fitting V to an upper expectile of Q over dataset actions,

$$\mathcal{L}_V = \mathbb{E}_{(s,a)\sim\mathcal{D}}\Big[\,L_2^\tau\big(\bar{Q}(s,a) - V(s)\big)\Big], \tag{7}$$

with $L_2^\tau(u) = \big|\tau - \mathbf{1}\{u<0\}\big|\,u^2$, and $\bar{Q}$ a target network updated by Polyak averaging (rate 0.005). The expectile $\tau = 0.7$ biases V toward the higher observed Q-values, which keeps it pessimistic about actions absent from the data. The Q-function regresses on the one-step Bellman target $y = r + \gamma\,(1 - \mathbf{1}_{\text{end}})\,V(s')$,

$$\mathcal{L}_Q = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\Big[\big(y - Q(s,a)\big)^2\Big], \tag{8}$$

with $\gamma = 0.995$, chosen so that terminal reward retains signal across the 50-step horizon ($0.995^{50} \approx 0.78$). The policy is extracted by advantage-weighted regression [13] with weights $w = \exp\big(\beta\,(\bar{Q}(s,a) - V(s))\big)$,

$$\mathcal{L}_\pi = -\,\mathbb{E}_{(s,a)\sim\mathcal{D}}\big[\, w\, \log \pi(a \mid s)\,\big], \tag{9}$$

with temperature $\beta = 3$ and w clipped at 100 for stability. All networks train with Adam (learning rate 3×10^{-4}), batch size 256, for 30,000 gradient steps. Every configuration runs with ten seeds, and at evaluation time the policy acts greedily. As secondary reference points we also train behavior cloning (BC), which imitates dataset actions without any reward and shows what the data alone induces, and an AWR variant of IQL without the expectile value function.

4 Experiments

Our experiments address three questions. **RQ1 (behavior):** holding data, algorithm, and evaluation fixed, does the choice of reward change the structure of the curriculum a policy teaches, and is the change consistent across learner models? **RQ2 (performance):** how do policies trained on each reward score on the other rewards' evaluation metrics? **RQ3 (faithfulness and robustness):** (a) does the preference-trained policy actually satisfy the judge it was distilled from, and (b) do the answers survive changes of student simulator and of offline RL algorithm?

Training grid and evaluation protocol. We train IQL on every combination of reward (quiz, scenario, preference), simulator (BKT, Coupled-BKT, DKT), and ten seeds, giving 90 primary policies (the BC and AWR reference points add another 150 runs), all with the configuration of Section 3.4 and no per-condition tuning. Evaluation is behavioral and in-distribution, meaning each policy is tested on the simulator it trained on and judged on how it teaches as well as what it scores. Each trained policy teaches 100 fresh simulated students from its training simulator, acting greedily for the full horizon, with episode seeds fixed given the policy seed so matched conditions face identical randomness. Each batch of 100 sessions is scored under all three reward functions and under the curriculum-structure metrics below.

Curriculum-structure metrics. These metrics, averaged over the 100 sessions, quantify how a policy teaches independently of attained mastery. *Concept entropy* is the Shannon entropy (nats) of the distribution of taught concepts within an episode, ranging from 0 (one concept for all 50 steps) to $\ln 16 \approx 2.77$ (uniform coverage). *Prerequisite respect* is the fraction of steps whose target concept had all prerequisites at mastery ≥ 0.5 when selected. *Mean difficulty* averages the selected difficulty over the episode. *First-hard step* is the index of the first hard-item selection. Episodes that never select a hard item are recorded at the horizon (50), so this quantity is right-censored and read qualitatively.

Statistical methodology. Headline comparisons are paired t-tests grouped into three families, each with a Holm correction that adjusts the threshold for the number of tests in the family and is reported as p_{Holm}. The families are a pooled family across all matched (simulator, seed) conditions ($n = 30$ pairs), a per-simulator family ($n = 10$), and an interaction family testing whether the preference-versus-quiz effect on DKT differs from its average on the two Bayesian simulators. We report exact p-values, paired Cohen's d as a standardized effect size, and mark non-significant results n.s.

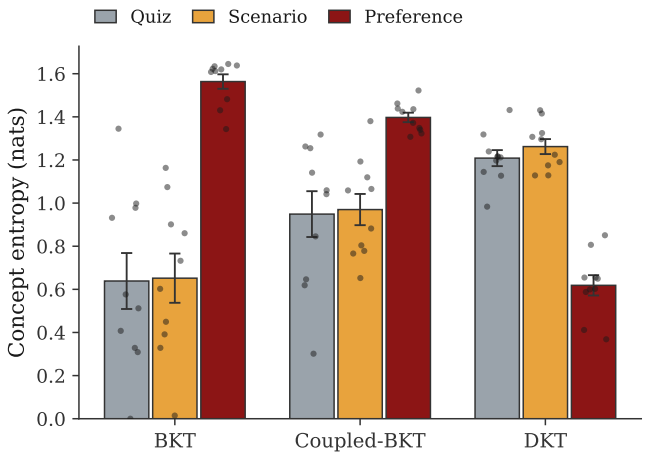

Figure 1: Concept entropy by training reward and simulator (dots are seeds, $n = 10$ per bar). The preference reward broadens the curriculum on the Bayesian simulators and narrows it on DKT.

Judge evaluation of trained policies. Learned reward models invite reward hacking, where a policy exploits blind spots of the model, scoring highly while producing sessions the judge would not actually prefer. We test for this by having the judge directly compare sessions from the trained policies. For each simulator and each opponent (quiz-trained and scenario-trained IQL), 25 session pairs of fresh rollouts across all ten seeds are judged under the identical rubric and both-order protocol used for label collection, and we report the preference policy's win rate among order-robust judgments.

5 Results

5.1 Reward choice reshapes teaching, and the direction depends on the learner model (RQ1)

Figure 1 and Table 1 show the main result. On the two Bayesian simulators, the preference-trained policy teaches more broadly than either hand-specified reward. Relative to quiz training, concept entropy rises by 0.93 nats on BKT ($d = 2.39$, $p_{\text{Holm}} = 7 \times 10^{-5}$) and 0.45 nats on Coupled-BKT ($d = 1.22$, $p_{\text{Holm}} = 0.008$). On DKT the effect reverses, and the preference policy is the narrowest (-0.59 nats, $d = -4.27$, $p_{\text{Holm}} < 10^{-4}$). The reward-by-simulator interaction is large ($d = -3.82$, $p < 10^{-4}$). Pooled across simulators, the opposing changes nearly cancel ($p = 0.054$, n.s.), so an aggregated analysis would miss both real effects. The two hand-specified rewards produce nearly identical behavior everywhere (entropy differences of 0.01 to 0.06 nats, all n.s.), so the key behavioral divide is between hand-specified and learned objectives.

The rest of the preference policy's behavior looks the same on every simulator. It respects prerequisites almost always (0.998 ± 0.001, the highest among the reward-trained policies), selects easier items (mean difficulty 1.336 ± 0.046 versus 1.878 to 1.970), and escalates to hard items later (first-hard step 28.8 versus 20.1 for quiz training), matching the rubric's calibrated-challenge criterion.

Table 1: Per-simulator results (IQL, in-distribution, mean ± SE over $n = 10$ seeds), with behavior cloning (BC) as a reward-free reference. Stars mark the preference-versus-quiz paired test within that simulator after Holm correction (*** $p_{\text{Holm}} < 0.001$, ** < 0.01; n.s. = not significant).

Simulator	Training	Quiz acc.	Scenario	Entropy
BKT	Quiz	0.344 ± 0.011	0.106 ± 0.007	0.639 ± 0.130
	Scenario	0.346 ± 0.009	0.106 ± 0.007	0.652 ± 0.114
	Pref.	0.443 ± 0.004***	0.172 ± 0.003***	1.564 ± 0.034***
	BC	0.336 ± 0.010	0.098 ± 0.008	0.520 ± 0.132
Coupled-BKT	Quiz	0.429 ± 0.009	0.171 ± 0.009	0.949 ± 0.106
	Scenario	0.429 ± 0.007	0.172 ± 0.008	0.970 ± 0.072
	Pref.	0.521 ± 0.002***	0.251 ± 0.002***	1.397 ± 0.022**
	BC	0.436 ± 0.009	0.175 ± 0.009	1.067 ± 0.100
DKT	Quiz	0.410 ± 0.007	0.152 ± 0.006	1.209 ± 0.037
	Scenario	0.410 ± 0.003	0.151 ± 0.003	1.262 ± 0.035
	Pref.	0.409 ± 0.003 n.s.	0.160 ± 0.003 n.s.	0.619 ± 0.047***
	BC	0.421 ± 0.003	0.158 ± 0.003	1.146 ± 0.047

5.2 The preference reward transfers where it broadens (RQ2)

Pooled over simulators, the preference-trained policy achieves the strongest performance on both objective metrics it was never trained on, with quiz accuracy 0.458 versus 0.394 for the quiz-trained policy ($d = 1.18$, $p_{\text{Holm}} < 10^{-5}$) and scenario success 0.194 versus 0.143 for the scenario-trained policy ($d = 1.37$, $p_{\text{Holm}} < 10^{-6}$). Disaggregation (Table 1, Figure 2) concentrates the transfer exactly on BKT and Coupled-BKT, where the preference reward also broadens the curriculum ($d = 3.1$–3.3, null on DKT).

The mechanism is visible in how the simulators score mastery. On the Bayesian simulators, a practiced concept reaches high mastery quickly while an unpracticed one stays near the guess rate. The quiz metric averages over all 48 cells, and every scenario needs all of its concepts at once, so narrow teaching leaves easy points on the table while broad teaching collects them. Teaching broadly is simply the better strategy for both hand-specified metrics on those learner models, and the preference reward, by pushing toward breadth, lands on that strategy while the metrics' own optimizers never find it. BC shows why. It matches the quiz- and scenario-trained policies on their own metrics everywhere (Table 1), so directly optimizing measured mastery gained nothing over imitating the data. The learned reward was the only signal that moved the policy somewhere new.

5.3 The judge endorses the result, including the reversal (RQ3a)

The head-to-head evaluation rules out reward hacking (Figure 3). The judge prefers the preference policy's sessions in 100% of order-robust comparisons on BKT and Coupled-BKT (78 of 78 across both opponents) and in 93% and 80% on DKT. The DKT numbers matter most here. On the one simulator where the preference policy teaches narrowly, the judge still prefers its sessions, so the reversal does not signal a broken pipeline. It reflects a faithful adaptation. For this judge, breadth is a means to good teaching rather than the goal itself, and the learned reward adapts what it asks of the policy to each learner model.

5.4 Robustness (RQ3b)

The findings do not depend on IQL specifically. The AWR variant, trained on the same relabeled data, reaches nearly identical preference-policy results (0.463 quiz accuracy, 0.196 scenario success,

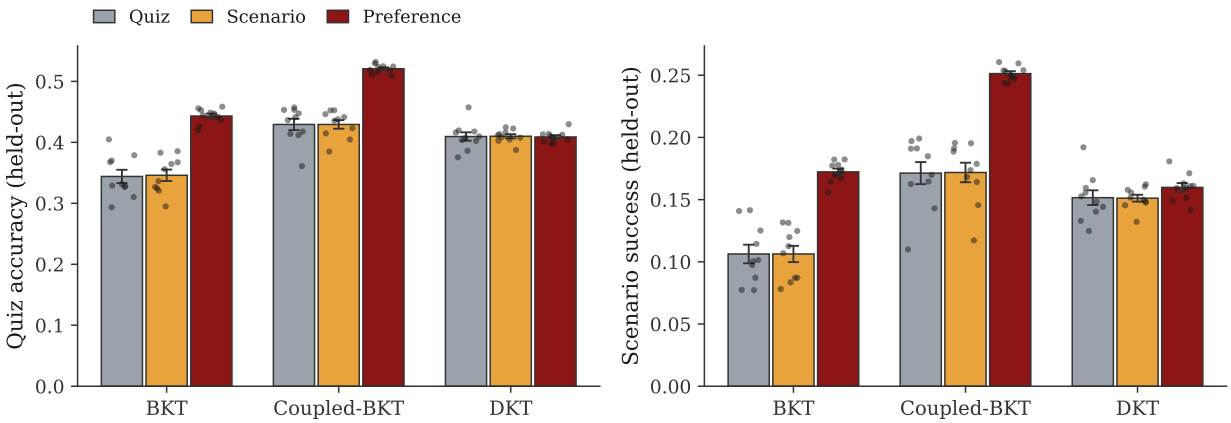

Figure 2: Held-out quiz accuracy (left) and scenario success (right) by training reward and simulator. The preference policy wins both metrics where it broadens the curriculum (BKT, Coupled-BKT) and ties on DKT, where it narrows.

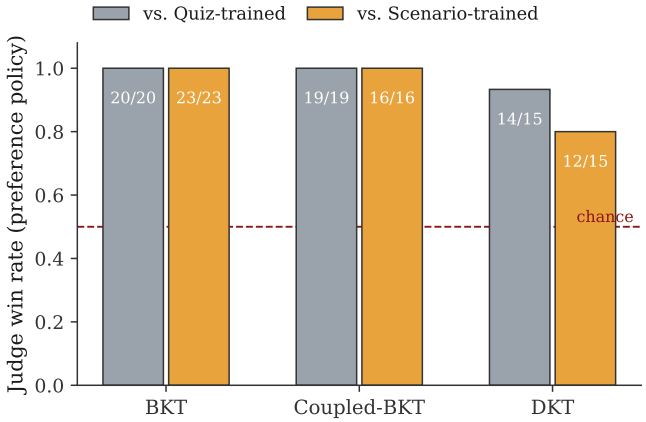

Figure 3: Judge preference for the preference policy's sessions over each specialized policy's, per simulator (25 pairs per cell, both-order judging, win rates among order-robust judgments).

1.225 pooled entropy, each within one standard error of IQL), and BC, which never sees the reward, shows none of the behavioral differences while matching attained reward (Section 5), confirming that the effects flow through the reward signal rather than the data. A behavioral probe also confirms the DKT simulator genuinely learned the prerequisite structure it was trained on, reproducing about 90% of the hand-coded transfer effect.

6 Discussion and Limitations

The reward is a design lever, and pooling can hide its effects. The reward alone moved curriculum breadth by 0.9 nats one way on BKT and 0.6 the other way on DKT, yet pooled analysis, a common default in simulator-based education research, would call it marginal. Behavioral effects of reward design interact with the learner model, so pooled claims should not be reported without disaggregation. In domains with no standardized assessment to inherit an objective from, the optimization target is itself a pedagogical decision with consequences for what students experience,

and policies can earn identical scores while teaching in completely different ways, so score alone cannot distinguish them.

What the preference channel did and did not do. The head-to-head evaluation of Section 5 already addresses reward hacking. We can also rule out rubric circularity. An earlier rubric ranked breadth first, whereas the present rubric is unordered and holistic, and the breadth effect persisted on the Bayesian simulators and reversed on DKT, so the policy tracks the judge's situational judgment rather than a keyword in the prompt. The results do not show that LLM judges discover good pedagogy. The judge's standard is whatever `claude-haiku-4-5` considers good teaching under our rubric, untested against human learning outcomes.

Why does the direction reverse on DKT? Section 5 establishes that the reversal is real and judge-endorsed but not why. One explanation is that on DKT every policy already teaches broadly (Table 1) because the learned dynamics diffuse credit across concepts, so additional breadth is cheap. With breadth saturated, the judge's calibrated-challenge criterion dominates, and the preference policy's gentler difficulty profile fits that reading.

Limitations. The main limitation is that all results are in simulation, and the three simulators, though structurally diverse, belong to one family. DKT is trained on Coupled-BKT data, and all three rungs share binary responses, stationary parameters, and the same difficulty model, so transfer to human learners is untested. Our prerequisite graph is itself a designed artifact, and a different graph could shift the results. One judge model under one rubric produced the labels, and a different judge or rubric would plausibly induce different behavior. The reward model reads a fixed summary of each session rather than the full sequence, so temporal structure reaches it only through aggregate features. The per-simulator tests rest on ten seeds, leaving smaller effects unresolved (the DKT scenario-versus-quiz entropy gap, for instance, sits at $p = 0.076$). Finally, the scope is one domain, sixteen concepts, a 50-step horizon, and an action space of concept-difficulty pairs.

7 Conclusion

This work asked whether the terminal reward changes how an offline-RL tutoring policy teaches, beyond how well it scores, and the answer is that reward design is a behavioral lever whose direction depends on the learner model. The same preference reward broadens teaching on the Bayesian simulators and narrows it on DKT, beats each hand-specified reward on its own metric where it broadens, and satisfies its judge everywhere. The reward function deserves explicit, audited design, evaluation should include behavioral measures, and simulator claims should be disaggregated by learner model. The clearest open directions are a move to human learners, sequence-level reward models, a wider set of judges and rubrics, and a mechanistic account of the DKT reversal.

Appendix: The Prerequisite Graph

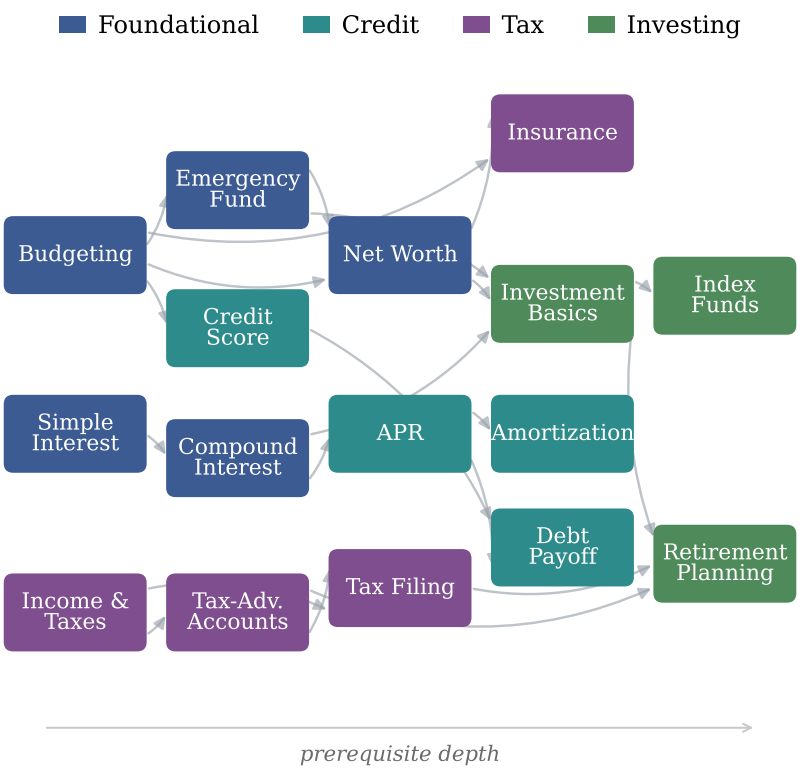

Figure A1: The 16-concept curriculum and its 21 prerequisite edges, colored by domain and arranged by prerequisite depth.

Appendix: The Judge Rubric

The system prompt below reproduces the complete rubric, typeset but otherwise word-for-word. The identical text is used for all three simulators' label collections and for the head-to-head evaluation. Criteria are deliberately unordered, and the judge returns a single character. Each pair is judged twice with presentation order swapped, and only order-robust verdicts train the reward models. Raw judge responses for both orders are archived with every label.

> *You are an experienced personal finance teacher comparing two tutoring sessions. In each, a tutor guided a beginner student through 50 practice items.*
>
> *Judge which session shows better teaching. Good teaching here means:*
>
> - *the student builds real mastery across several concepts, not just one*
> - *foundational concepts come before the concepts that build on them*
> - *challenge fits the student: difficulty rises as mastery grows, and the student succeeds often enough to stay motivated*
>
> *Respond with EXACTLY ONE CHARACTER: "A" if Session A shows better teaching, "B" if Session B does. No other text.*

Contributions of Team Members

The division of labor largely followed the milestone plan, and Daniel and Stella contributed equally to the poster and the paper.

- **Daniel Argento:** took the lead on the code implementation used to run the experiments, and led the methods, experiments, and results sections of the paper.
- **Stella Wu:** constructed the preference reward pipeline, in which an LLM judge reviewed trajectories, helped scope the experimental trials and the adaptations made along the way, and led the related work and discussion sections of the paper.

Several structural changes followed the milestone. The single BKT simulator became a three-rung ladder (BKT, Coupled-BKT, and DKT) so that reward-design claims could be tested for robustness to the learner model, and the preference reward, only stubbed in code at the milestone, was fully realized through an LLM judge with a per-simulator Bradley–Terry model and both-order judging. The datasets grew from 600 to 10,002 trajectories per simulator, training moved from Monte Carlo returns to the full Bellman backup, and the grid expanded to ten seeds. These changes also sharpened the finding, since the key divide proved to be between hand-specified and learned rewards rather than scenario versus quiz, and the behavioral effect turned out to be moderated by the learner model rather than robust to it.

AI Use Disclosure

Our team used Claude (Anthropic) in several capacities throughout this project. Specifically, we used it to assist with LaTeX formatting and document structure, generate and refine matplotlib/visualization code for figures and data aggregation, and produce boilerplate scaffolding for ideas we subsequently developed independently. We also consulted Claude to clarify conceptual differences between knowledge tracing models (BKT, coupled BKT, and DKT) as a supplementary reference alongside course materials and papers. Finally, we used Claude to assist in crafting and iterating on prompts used in our pipeline. All core algorithmic contributions, including the IQL training loop, reward design, environment logic, and experimental analysis, were developed independently by the team.

References

[1] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.

[2] Jonathan Bassen, Bharathan Balaji, Michael Schaarschmidt, Candace Thille, Jay Painter, Dawn Zimmaro, Alex Games, Ethan Fast, and John C. Mitchell. Reinforcement learning for the adaptive scheduling of educational activities. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2020.

[3] Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[4] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[5] Albert T. Corbett and John R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1995.

[6] Shayan Doroudi, Vincent Aleven, and Emma Brunskill. Where's the reward? a review of reinforcement learning for instructional sequencing. *International Journal of Artificial Intelligence in Education*, 29:568–620, 2019.

[7] Cheryl I. Johnson and Jennifer L. Solberg. Designing an adaptive instructional system for financial literacy training. In *HCI International 2023 – Late Breaking Papers*, volume 14060 of *Lecture Notes in Computer Science*, pages 116–127. Springer, 2023.

[8] Brian W. Junker and Klaas Sijtsma. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25 (3):258–272, 2001.

[9] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022. arXiv:2110.06169.

[10] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[11] Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popović. Offline policy evaluation across representations with applications to educational games. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems*, 2014.

[12] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[13] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

[14] 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*, volume 28, 2015.

[15] Anna N. Rafferty, Emma Brunskill, Thomas L. Griffiths, and Patrick Shafto. Faster teaching via POMDP planning. *Cognitive Science*, 40(6):1290–1332, 2016.

[16] John Tan. Discovering metacognitive approaches to personal finance with intelligent tutoring systems, 2018. Manuscript, Georgia Institute of Technology.

[17] Kurt VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.