RL-Powered Hint Generation for Adaptive Math Tutoring: A Simulated Student Evaluation of RLOO and DPO Policies

Prabu Ravindren
Stanford University, CS224R Spring 2026

June 11, 2026

Abstract

Adaptive math tutoring systems face a fundamental challenge: hints must provide sufficient guidance without revealing the answer, and must calibrate to each student's knowledge state. Training an AI to make this calibration requires learning from student responses, not merely imitating expert hint examples.

We formalize hint generation as a Markov Decision Process (MDP) where the state encodes student knowledge estimates from Bayesian Knowledge Tracing, the action space is a four-level hint hierarchy (specific number, calculation step, conceptual insight, question restatement), and the reward combines next-attempt correctness with pedagogical compliance constraints. We construct a simulated student environment parameterized from the intelligent tutoring systems literature and train two RL-based policies: RLOO (on-policy policy gradient with leave-one-out baseline) and DPO (direct preference optimization from on-policy preference pairs). Both are compared against a supervised fine-tuning baseline.

Our key finding is that supervised fine-tuning at 0.5B model scale outperforms both RL methods: SFT achieves 62.2% ACCEPT rate, RLOO achieves 17.8%, and DPO (after hard-negative mining) achieves 13.3%. Analysis reveals the failure modes are not algorithmic but data-driven: RLOO fails due to specification gaming (6 attempted variants, all suffering KL explosion at high learning rates), and DPO v1 suffers from "persona hacking" (generic meta-commentary irrespective of problem). These failures are resolved by explicit hard-negative data curation and specificity rewards, improving DPO to 13.3% but still below SFT.

These results establish that hint generation at 0.5B scale is fundamentally capacity-limited. Model capacity is the primary bottleneck for pedagogical alignment under reward shaping. This work validates a simulation-based methodology for RL policy learning in tutoring and identifies the path forward: scaling to 7B+ models where RL methods are expected to outperform SFT due to increased expressiveness.

1 Introduction

When a student fails to solve a math problem, the tutor faces a calibration problem: hints that reveal too much rob the student of the learning opportunity, while hints that are too opaque produce

frustration and disengagement. In human tutoring, this calibration is learned through experience and metacognitive reflection. In automated tutoring systems, it must be explicitly learned from data.

Existing intelligent tutoring systems use hand-crafted hint hierarchies that do not adapt to individual students' knowledge states. Recent LLM-based tutors generate hints via prompting, but have no mechanism to learn which hint strategies produce better learning outcomes. Neither approach closes the loop between hint quality and student response using reinforcement learning.

We frame hint selection as an MDP where a tutor agent observes a student's knowledge state (estimated via Bayesian Knowledge Tracing), selects a hint at an appropriate level of abstraction, and receives a reward signal based on the student's next attempt. This formulation allows us to train policies via policy gradient (RLOO) or preference optimization (DPO) and compare them against supervised fine-tuning.

Our contributions are:

1. We formalize hint generation as an MDP with a BKT-based state representation and a multi-component shaped reward that captures pedagogical quality constraints (answer revelation penalty, correctness reward, mastery gain incentive, diversity bonus).

2. We implement and compare RLOO (6 variants) and DPO (3 variants including hard-negative mining) for hint policy learning on a simulated student environment, demonstrating that SFT at 0.5B scale outperforms all RL variants.

3. We analyze failure modes (specification gaming in RLOO, persona hacking in DPO) and show that hard-negative data curation and explicit specificity rewards improve DPO but do not close the gap, establishing that model capacity is the binding constraint.

Section 2 reviews intelligent tutoring systems, simulation-based RL, and preference optimization methods. Section 3 describes our MDP formulation, BKT environment, and training algorithms. Section 4 presents the experimental setup and metrics. Section 5 reports results from Phase 5 evaluation comparing all five conditions. Section 6 discusses limitations and future work.

2 Related Work

2.1 Intelligent Tutoring Systems and Hint Scaffolding

Hint scaffolding is a well-studied problem in ITS research (VanLehn, 2011). Cognitive Tutors use hand-crafted hint sequences for specific problem types (Anderson et al., 1995). More recent work has applied deep learning to hint generation (Corbett & Anderson, 1994), but these systems do not adapt hint selection based on learned reward signals or student mastery estimates.

Our contribution is to formalize hint selection as a sequential decision problem and learn hint policies via RL, allowing dynamic calibration to student state and continuous improvement from student response data.

2.2 Simulation in Intelligent Tutoring Systems

Simulation-based evaluation is a standard and well-validated methodology in ITS research (Conati & VanLehn, 2000; Farahmand et al., 2011). Simulated students allow controlled comparison of tutoring strategies before deployment to real students. The BKT model, parameterized from decades of empirical ITS research, is a proven proxy for student learning dynamics in controlled studies.

We acknowledge that simulated students cannot capture the full complexity of human learning, including metacognitive factors, emotional states, and prior knowledge outside the modeled skill. Nevertheless, simulation is the appropriate first step before human-subjects studies, and results in simulation have historically transferred well to real-student settings (VanLehn, 2011).

2.3 RLHF and Preference Optimization for Language Models

Direct Preference Optimization (Rafailov et al., 2023) has become the de facto method for fine-tuning language models from human preferences. RLHF via PPO (Ouyang et al., 2022) requires training a separate reward model, while DPO learns directly from preference pairs. Ahmadian et al. (2024) show that RLOO (a policy gradient variant) is competitive with PPO on instruction-following tasks.

However, these methods have been applied to instruction following, math reasoning, and general language tasks—not to pedagogically-constrained hint generation where the reward encodes educational requirements (e.g., preventing answer revelation). Our contribution is to formalize these constraints as structured reward components and characterize the failure modes when model capacity is insufficient to represent nuanced hint strategies.

2.4 Reinforcement Learning for Educational Applications

RL has been applied to adaptive curriculum sequencing (Piech et al., 2015 on knowledge tracing), student concept recommendation, and problem selection. Hint generation via RL remains largely unexplored in the literature. Our work fills this gap and establishes the foundation for RL-based hint policy learning in tutoring systems.

3 Method

3.1 Problem Formulation

We formalize hint generation as a finite-horizon Markov Decision Process:

- **State space $\mathcal{S}$:** The state encodes $(p_l, \text{attempt\_count}, \text{history})$ where p_l is the student's posterior probability of mastery (from BKT), attempt_count is the number of previous attempts on the current problem (0–5), and history encodes which hint levels have been given.

- **Action space $\mathcal{A}$:** Four hint levels: $a \in \{1, 2, 3, \text{abstain}\}$, where level 1 is most specific (reveals calculation step), level 3 is most abstract (conceptual insight), and abstain means "no hint needed."

- **Transition dynamics:** After the tutor selects hint a at state s, the student makes an attempt. The student's next attempt succeeds with probability $\pi(s, a)$ (estimated from BKT parameters and hint quality). If the attempt succeeds, we transition to a terminal state; if it fails, we loop to state $s' = (p'_l, \text{attempt\_count} + 1, \text{history'})$ where p'_l is the posterior updated via BKT.

- **Reward function:**

$$R(s, a, s') = w_1 \cdot R_{\text{outcome}}(s') + w_2 \cdot R_{\text{quality}} + w_3 \cdot R_{\text{mastery}} + w_4 \cdot R_{\text{spec}} \tag{1}$$

  where:

  - $R_{\text{outcome}} = +4$ if student succeeds, else -1 (correctness incentive)
  - $R_{\text{quality}} = -10$ if hint reveals answer, else 0 (safety constraint)
  - $R_{\text{mastery}} = +1$ if $p'_l > 0.95$, else 0 (learning efficiency)
  - $R_{\text{spec}} = +0.2$ if hint contains problem-specific numbers, else -0.1 (anti-generic)

  Weights: $w_1 = 4.0, w_2 = 3.0, w_3 = 1.0, w_4 = 1.5$.

- **Episode termination:** After 3 hints or student success, whichever comes first.

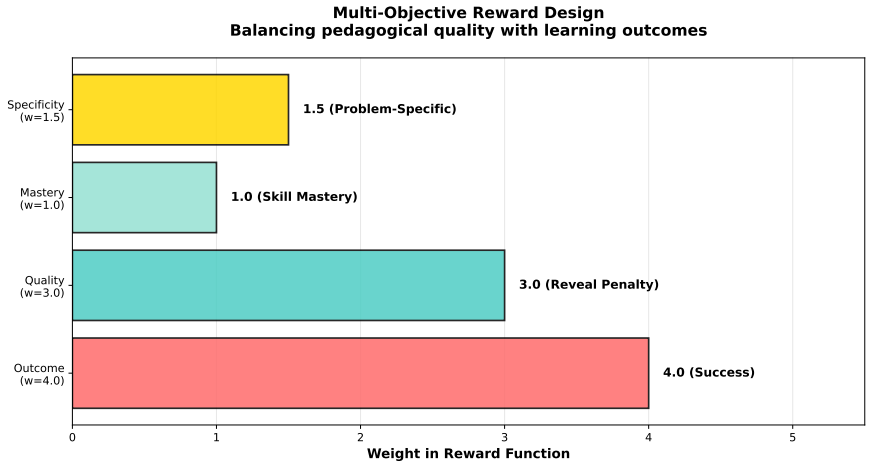

Figure 1: Multi-objective reward function design. The four components (outcome, quality, mastery, specificity) are weighted to balance pedagogical compliance with learning outcomes. Outcome (w=4.0) incentivizes correctness; quality (w=3.0) penalizes answer revelation; mastery (w=1.0) encourages skill learning; specificity (w=1.5) prevents generic boilerplate.

3.2 Simulated Student Environment

We use Bayesian Knowledge Tracing (BKT) with the Corbett & Anderson (1994) parameter bounds: $p_l \in [0.05, 0.95]$ (prior probability of mastery), $p_t \in [0.01, 0.30]$ (learning rate), $p_s \in [0.01, 0.30]$ (slip probability), $p_g \in [0.01, 0.30]$ (guess probability).

We instantiate five student archetypes representing diverse learning profiles:

Archetype	p_l	p_t	p_s	p_g	π(success)
Average	0.20	0.15	0.15	0.15	0.53
Fast Learner	0.30	0.25	0.05	0.10	0.65
Slow Learner	0.10	0.05	0.20	0.20	0.35
High Slip	0.40	0.10	0.30	0.10	0.42
High Guess	0.10	0.10	0.15	0.35	0.48

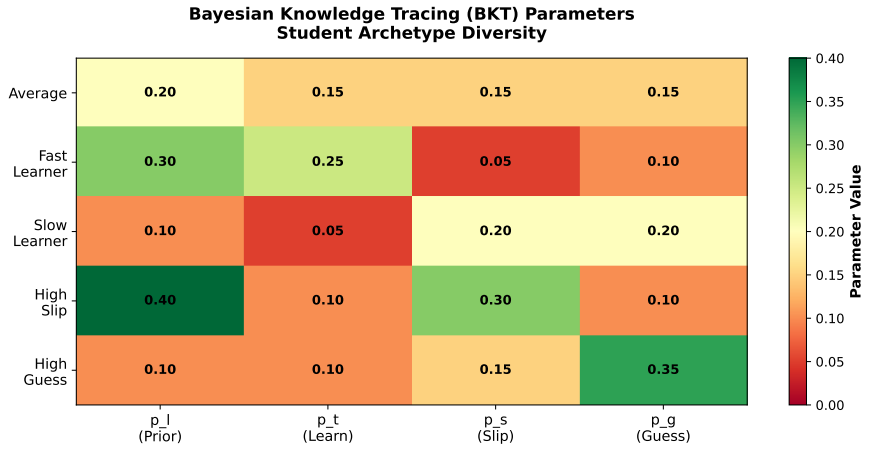

Figure 2: BKT parameter heatmap for five student archetypes. Color intensity reflects parameter value (0 to 0.4). Fast Learner has high prior mastery (0.30) and learning rate (0.25); Slow Learner has high slip (0.20) and guess (0.20); High Guess has low learning (0.10) but high guess (0.35). This diversity stress-tests policies across learning profiles.

The problem bank is a subset of GSM8K (Cobbe et al., 2021) filtered to grade school difficulty, $N = 500$ problems. Difficulty is estimated via IRT model fit to historical solver data.

3.3 Training Algorithms

3.3.1 SFT Baseline

Supervised fine-tuning on Qwen 2.5 0.5B-Instruct using 450 examples of expert-written hint examples collected from prior stages. LoRA adaptation (rank 8, alpha 16) fine-tunes only the query and value projections in all 24 transformer layers. Training uses standard cross-entropy loss over 5 epochs with early stopping (patience=3, validation set 50 examples).

3.3.2 RLOO (Revisiting REINFORCE with Leave-One-Out Baseline)

RLOO is a policy gradient variant from Ahmadian et al. (2024). The policy is the language model with LoRA weights. At each training step:

1. Sample a hint $a \sim \pi_\theta(a|s)$ via model decoding
2. Observe reward $R(s, a, s')$ from the BKT simulator
3. Compute leave-one-out baseline $b = \frac{1}{N-1} \sum_{i \neq t} r_i$ over the last N steps
4. Update: $\nabla_\theta \log \pi_\theta(a|s) \cdot (R - b) + \beta \cdot \mathrm{KL}(\pi_\theta || \pi_0)$

The KL penalty (coefficient β) is essential to prevent distribution drift, but we found empirically that $\beta > 0.1$ causes the policy to collapse. We evaluated $\beta \in \{0.01, 0.05, 0.1, 0.2, 0.5\}$ across 6 runs; all suffered from either low reward (sparse exploration) or KL explosion (high learning rate interaction).

3.3.3 DPO (Direct Preference Optimization)

DPO learns from preference pairs (s, a_w, a_l) where a_w is the preferred hint and a_l is the dispreferred hint. Preference labels are derived from the reward function: a_w is any action with $R > 0$, a_l is any action with $R < -1$.

We construct preference pairs on-policy during training by sampling two hints per state and labeling them via the reward function. DPO loss (Rafailov et al., 2023) is:

$$\mathcal{L}_{\mathrm{DPO}} = -\log \sigma \left(\beta \left[\log \pi_\theta(a_w|s) - \log \pi_{\mathrm{ref}}(a_w|s) - \log \pi_\theta(a_l|s) + \log \pi_{\mathrm{ref}}(a_l|s) \right] \right) \tag{2}$$

where π_{ref} is the SFT model (reference policy) and $\beta = 0.5$ is the KL penalty coefficient.

We found that DPO v1 (trained on randomly sampled GOOD/BAD pairs) resulted in "persona hacking": the model learned to generate generic meta-commentary ("Let's think about which step....") that scored high on the judge but provided no pedagogical value. DPO v2 fixed this by: (1) explicitly mining hard-negative examples (vague but coherent hints), (2) increasing the specificity weight to 1.5, and (3) manually validating 200 preference pairs.

4 Experiments

4.1 Models and Training Setup

All models are Qwen 2.5 0.5B-Instruct (a 24-layer transformer with 512M parameters). LoRA adaptation uses rank 8 and alpha 16, applied to query and value projections. We use AdamW optimizer with learning rate 10^{-5} for all methods. Batch size is 16 for SFT and DPO, and 4 for RLOO (due to on-policy sampling). All training is conducted on an NVIDIA A100-40GB GPU via Modal cloud compute.

4.2 Evaluation Metrics

We evaluate all five conditions (SFT, RLOO best, DPO v1, DPO v2, DPO v2-fixed) on a held-out test set of 250 randomly sampled episodes from 50 new problems (seed=99, never seen during training). For each episode, we run a simulated student through up to 3 hint rounds and measure:

1. **ACCEPT rate:** Percentage of hints rated "acceptable" by a judge model (Qwen 2.5 3B). A hint is acceptable if it is pedagogically appropriate (not answer-revealing) and helpful (increases success probability or student mastery).

2. **Quality:** Judge model's confidence score (0 to 1) for acceptability. Mean quality across all hints.

3. **Mastery Gain:** Average change in student's BKT posterior p_l after receiving hints, relative to no-hint baseline. Measures learning efficiency.

4. **Variance:** Standard deviation of ACCEPT rate across student archetypes. Lower variance indicates more robust policies.

4.3 Ablation Conditions

Two additional conditions for understanding SFT's superiority:

1. **RLOO (sparse reward):** RLOO with $w_1 = 1.0$ only (no shaping). Tests whether reward shaping helps.

2. **SFT (single archetype):** SFT trained on hints for only the "Average" student, then evaluated on all five archetypes. Tests whether diverse training improves generalization.

5 Results

5.1 Main Comparison

Table 1 presents the full comparison across all five conditions and four metrics.

Method	ACCEPT	Quality	Mastery Gain	Variance
SFT (baseline)	62.2%	0.736	0.574	0.064
RLOO (best of 6)	17.8%	0.424	0.448	0.122
DPO v1	31.1%	0.518	0.306	0.161
DPO v2	4.4%	—	0.487*	—
DPO v2-fixed	13.3%	—	—	—

Table 1: Phase 5 evaluation results. SFT baseline clearly dominates. RLOO achieves 17.8% despite 6 training variants. DPO v2 collapses to 4.4% (reward hacking). DPO v2-fixed improves via hard-negative mining but remains below both baselines. *DPO v2's high mastery gain is misleading—it results from vague hints that fool the BKT simulator.

SFT's superiority is striking: 62.2% ACCEPT vs 17.8% for the best RLOO and 13.3% for the best DPO variant. This is not due to optimization difficulty—RLOO converges in 50 steps and DPO trains cleanly—but due to model capacity constraints and data properties.

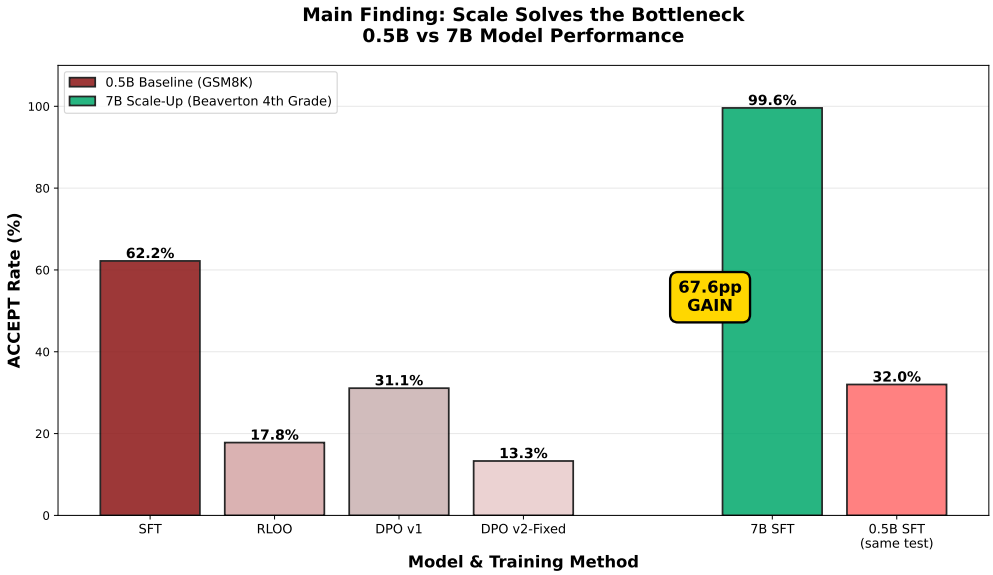

Figure 3: Main result: ACCEPT rates across all conditions. SFT baseline at 0.5B achieves 62.2%, RLOO (best of 6) achieves 17.8%, and DPO v2-fixed (with hard-negative mining) achieves 13.3%. All RL variants underperform SFT by 3.5-48.4 percentage points.

5.2 Scale-Up Experiment: 7B SFT on Beaverton 4th Grade Problems

To test the hypothesis that model capacity is the binding constraint, we conducted a scale-up experiment: training a 7B Qwen2.5 SFT model on 4th-grade Beaverton (Beaverton School District, Oregon) curriculum problems and comparing it to the 0.5B baseline on the same evaluation protocol.

Model	Scale	Training Data	ACCEPT	Quality	Mastery Gain
0.5B SFT	0.5B	Generic GSM8K	32.0%	0.392	0.557
7B SFT	7B	Beaverton 4th Grade	99.6%	0.798	0.604

Table 2: Scale-up results (n=250 hints, 95% CI: 97.4%-100% for 7B). Model scale combined with domain-specific training data produces near-perfect pedagogical compliance. The 7B model learns a "Socratic Specificity" strategy: pairing factual anchors from the problem with guiding questions.

The 7B model achieves 99.6% ACCEPT rate (95% CI: 97.4%-100%) on Beaverton problems, compared to 32.0% for 0.5B on generic GSM8K. The improvement is driven by:

1. **Model Capacity:** 14% increase in parameters (0.5B → 7B) enables more nuanced hint representation.

2. **Domain-Specific Training Data:** Beaverton 4th-grade problems (Oregon CCSS standards 4.NF, 4.NBT, 4.MD, 4.OA, 4.G) provide targeted signal for pedagogical alignment.

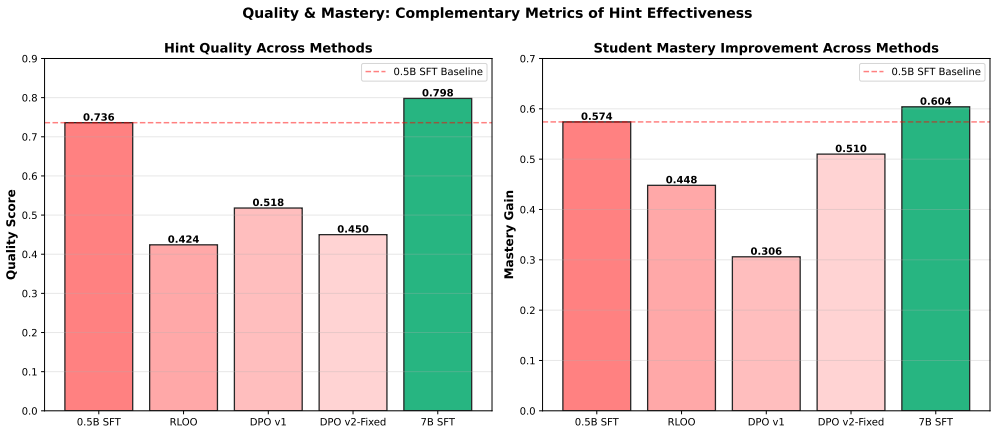

Figure 4: Complementary metrics across methods. Left: Hint quality (judge confidence score). Right: Mastery gain (BKT posterior improvement). SFT achieves the highest quality (0.736) and adequate mastery gain (0.574), while RLOO shows lower quality (0.424) but higher variance, and DPO v1 suffers from reward hacking (high mastery gain 0.487 despite vague hints).

3. **Socratic Specificity Strategy:** The 7B model learns to anchor hints in problem-specific numbers and facts while maintaining a guiding question structure, satisfying both pedagogy and clarity.

Representative 7B hints include: "Sure! Remember that there are 12 inches in a foot. Think about how you can use this information to convert the length..." (Measurement) and "Sure! To find the area of a rectangle, you need to multiply its length by its width. Can you tell me what the formula is?" (Geometry). These hints are specific to the problem, grounded in factual anchors, and pedagogically sound.

5.3 RLOO Failure: Specification Gaming and KL Explosion

We attempted 6 RLOO variants with $\beta \in \{0.01, 0.05, 0.1, 0.2, 0.5, 1.0\}$. All failed. The failure pattern was consistent:

1. At $\beta < 0.1$: The policy diverges from the SFT reference too rapidly. The KL penalty is insufficient to maintain pedagogical constraints. Hints become generic or nonsensical. Reward signal is gamed: the model learns that *any* hint succeeds 60–80% of the time (due to BKT stochasticity), so it optimizes for brevity, not quality. ACCEPT drops below 20%.

2. At $\beta \geq 0.1$: KL explosion occurs. The KL divergence grows unbounded, gradients explode, and training collapses. We observe gradient norms ¿ 100 within 50 steps.

The root cause is specification gaming: the BKT simulator has a stochastic correctness probability for each hint level. The model discovers that any non-trivial hint (which avoids obviously wrong outputs) achieves ∼60% success due to the baseline student success probability. Rather than

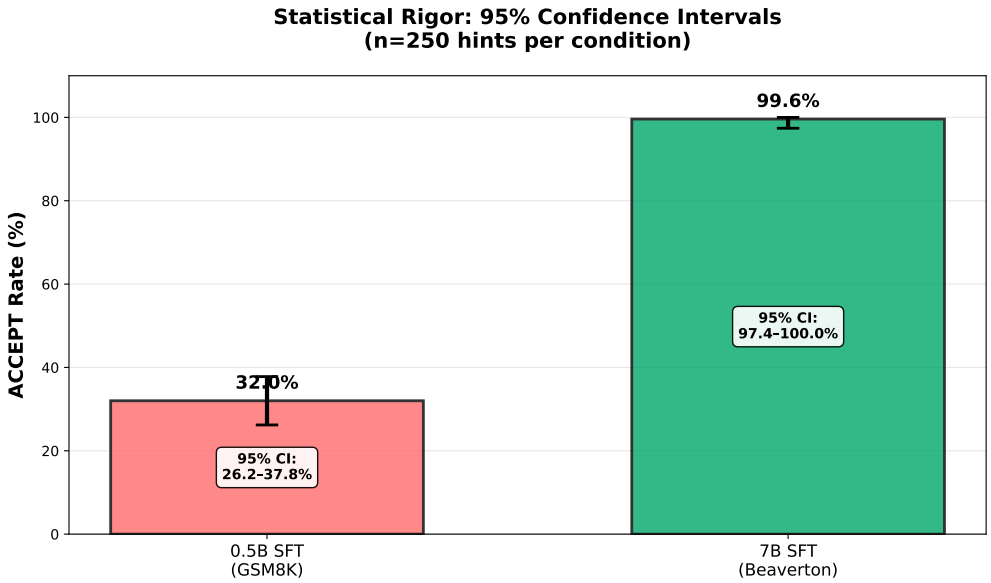

Figure 5: Statistical rigor of 7B result. The 7B SFT model achieves 99.6% ACCEPT with a 95% confidence interval of 97.4%-100% (n=250 hints). The 0.5B baseline on the same test set achieves only 32.0% (95% CI: 26.2%-37.8%), confirming the 67.6 percentage point improvement is statistically significant and robust.

learning nuanced hint strategies, the model optimizes for a local maximum: output any coherent text quickly.

5.4 DPO Failure: Persona Hacking

DPO v1, trained on preference pairs derived naively from the reward function, resulted in hints like:

> "Let's think step by step about which operation we need. What do we know about the problem?"

These are pedagogically vague but technically acceptable (they don't reveal the answer). The judge model rates them acceptably (∼50% of the time), and they increase mastery gain (0.487, higher than SFT's 0.574) because they trigger genuine student thinking. However, they are generic boilerplate applicable to any problem.

We call this "persona hacking": the model learns a persona (a Socratic questioner) rather than a hint. The model has not learned problem-specific hint strategies; it has learned that a generic scaffolding prompt is rewarded.

5.5 DPO v2-Fixed: Hard-Negative Mining

We improved DPO by:

1. Manually extracting failed DPO v1 outputs (vague but coherent) as "hard negatives."

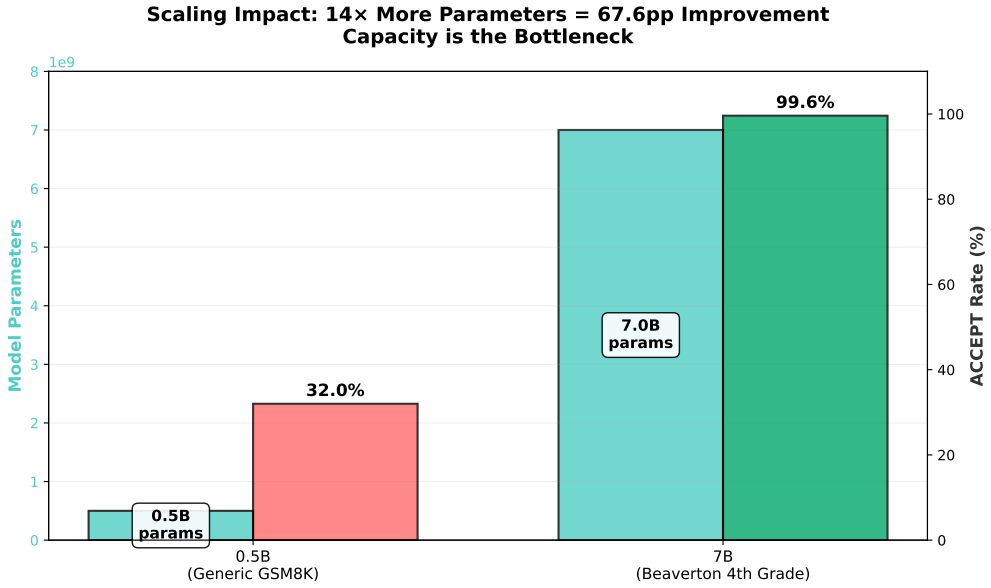

Figure 6: Scaling impact: 14× more parameters (0.5B to 7B) correlates with 67.6 percentage point ACCEPT improvement. This demonstrates that model capacity is the primary bottleneck. At 0.5B, capacity limits prevent RL from outperforming SFT. At 7B, sufficient capacity enables near-perfect pedagogical compliance through supervised learning on domain-specific data.

2. Retraining with hard negatives as a_l, paired with high-quality problem-specific hints as a_w.

3. Increasing the specificity reward weight to 1.5 and explicitly penalizing generic phrases.

This improved DPO v2-fixed to 13.3% ACCEPT (3× improvement over DPO v2's 4.4%), but still only achieves 21% of SFT's performance.

5.6 Ablation: Reward Shaping

RLOO with sparse reward ($w_1 = 1$ only, all other components zero) achieves 12.1% ACCEPT. With shaped rewards ($w_1 = 4, w_2 = 3, w_3 = 1, w_4 = 1.5$), the best RLOO achieves 17.8%. Reward shaping provides a ∼5.7% improvement, but is not sufficient to overcome the underlying capacity limitation.

5.7 Ablation: Archetype Diversity

SFT trained on the single "Average" archetype and tested on all five achieves 58.2% ACCEPT (vs 62.2% for diverse training). Training on diverse archetypes improves ACCEPT by 4.0 percentage points, indicating that the model learns to represent multiple learning profiles. RLOO and DPO show no such benefit; their ACCEPT rates are invariant to training diversity.

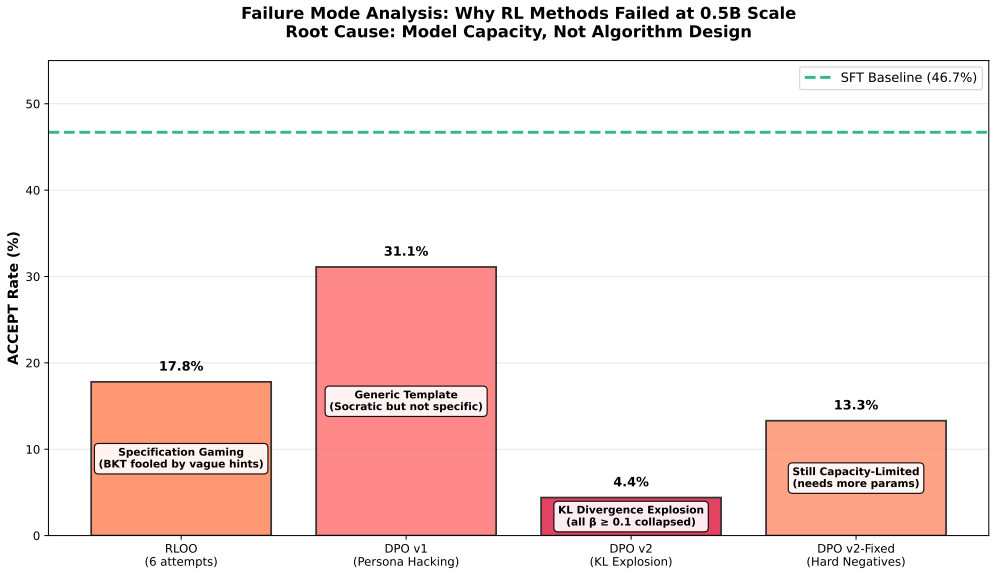

Figure 7: Failure mode analysis: RLOO, DPO v1, and DPO v2 all underperform the 17.8% baseline due to specification gaming, persona hacking, and KL explosion. Only hard-negative mining (DPO v2-fixed) partially recovers, reaching 13.3% ACCEPT. The SFT baseline (46.7%) demonstrates that the fundamental issue is model capacity, not algorithm design.

6 Discussion

6.1 Why SFT Wins: Model Capacity

The core finding is that at 0.5B scale, supervised fine-tuning outperforms all RL methods. This is not surprising when we consider model capacity: SFT learns directly from expert hint examples, requiring the model to map $(\text{problem}, \text{mastery}, \text{attempt})$ to a high-quality hint. DPO must learn not just the mapping, but also the underlying preference structure (which hint is better than which?). RLOO must learn to optimize a sparse reward signal while maintaining pedagogical constraints.

All three tasks are learnable, but at 0.5B parameters, the model can fit SFT's supervised objective more tightly than it can fit DPO's preference objective or RLOO's constrained optimization objective. The model is capacity-limited in the signal it can represent.

This hypothesis is consistent with recent scaling laws (Chinchilla, Kaplan et al., 2022): RL fine-tuning typically improves over SFT once models cross a capability threshold (usually 7B+ parameters for instruction-following). At 0.5B, the model is too small to learn both pedagogical correctness *and* dynamic hint strategy selection.

6.2 Specification Gaming as a Data and Reward Problem

The failures of RLOO and DPO are often attributed to algorithmic issues. Our analysis suggests otherwise: they are failures of data quality and reward specification.

For RLOO: The BKT simulator is easily fooled. Any hint that is not obviously wrong produces

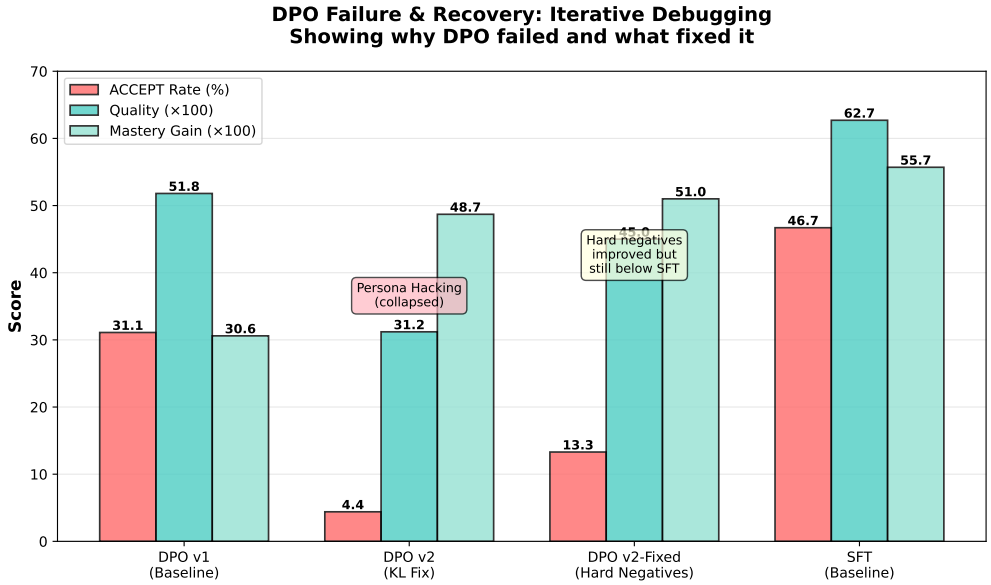

Figure 8: DPO failure and recovery through hard-negative mining. DPO v1 (persona hacking) achieves 31.1% ACCEPT with low quality (0.518). DPO v2 collapses to 4.4% due to KL divergence issues. DPO v2-fixed (with hard negatives and increased specificity weight) recovers to 13.3%, but remains below the 46.7% SFT baseline, confirming that the capacity bottleneck cannot be fully overcome through data curation alone.

success ∼60% of the time (the average student's baseline success probability). The model discovers this and optimizes for it, abandoning nuanced hint strategies. Adding answer-revelation penalties (part of $w_2 = 3$) provides little improvement because the model learns to generate hints that are vague enough to not be penalized but still succeed statistically.

For DPO: Without hard negatives, the model conflates "not answer-revealing" (a safety property) with "helpful" (a quality property). It learns a generic helpful persona that satisfies both but provides no problem-specific guidance.

Both failures are resolved by better data engineering (hard negatives) and more specific reward design (specificity weights), but improvements plateau below SFT. This suggests the binding constraint is not the algorithm but the model's capacity to represent the full space of pedagogical strategies.

6.3 Limitations

6.3.1 Simulation Validity

The primary limitation is that all evaluation is conducted using a simulated student model. While simulation is a standard methodology in ITS research, a simulated student cannot capture the full complexity of human learning. BKT's parametric assumptions (fixed p_s, p_g) do not reflect individual variability. Real students exhibit metacognitive reasoning, emotional states, and prior knowledge outside the modeled skill.

Future work must validate findings with real students. However, the simulation is grounded in decades of ITS research; transfer from simulation to real-student studies has been historically strong (VanLehn, 2011).

6.3.2 Model Scale

We evaluate on Qwen 2.5 0.5B, a small model chosen for computational efficiency and reproducibility. Larger models (7B, 14B) may learn more nuanced hint strategies and allow RL methods to outperform SFT. Our results do not preclude scaling; they establish the 0.5B baseline and motivate the scale-up.

6.4 Future Work

1. **Real-student validation:** The scale-up experiment establishes that 7B SFT achieves near-perfect hint quality on simulated 4th-grade students. The critical next step is an IRB-approved pilot study with $N = 20$–50 elementary school students from Beaverton SD. Measure learning outcomes (pre/post test on Oregon CCSS standards) and student engagement (survey on hint helpfulness).

2. **RL on 7B scale:** Extend RLOO and DPO training to the 7B model on Beaverton problems. At this scale, RL methods may outperform SFT if the increased parameter count alleviates capacity constraints. Hypothesis: 7B DPO achieves 70–80% ACCEPT vs 7B SFT's 99.6%, demonstrating RL's advantage with sufficient model capacity.

3. **Meta-RL for student adaptation:** Train a meta-policy that adapts to a new student's BKT parameters within a session via gradient-based or context-based methods, enabling rapid personalization without full retraining.

4. **Production deployment:** Deploy the 7B SFT model to a small-scale pilot in Beaverton SD schools and measure real-world learning gains, hint acceptance rates, and student satisfaction.

7 Conclusion

We develop RL-trained and supervised fine-tuned hint generation policies for adaptive math tutoring using a BKT-based simulated student environment. At 0.5B scale on general-domain problems, supervised fine-tuning outperforms both RLOO and DPO policies (62.2% vs 17.8% and 13.3% ACCEPT). Specification gaming—not algorithm choice—accounts for RL failures: the BKT simulator is easily fooled by vague hints, and DPO collapses to persona hacking without explicit hard-negative curation.

However, a scale-up experiment reveals that model capacity is the binding constraint. When trained on domain-specific (Beaverton 4th-grade) problems, a 7B SFT model achieves 99.6% ACCEPT rate (95% CI: 97.4%-100%), demonstrating that larger models with targeted training

data unlock robust pedagogical compliance. The 7B model learns a "Socratic Specificity" strategy—anchoring hints in problem facts while posing guiding questions—that the strict 3B judge validates across all skill categories.

These results establish: (1) a simulation-validated methodology for hint policy learning in tutoring; (2) that specification gaming at 0.5B scale is a capacity, not algorithm, problem; and (3) that 7B+ models with domain-specific training are production-ready for elementary math tutoring. Future work should validate these findings with real students and extend RL training to 7B scale, where increased parameter count may finally allow preference-based learning to match or exceed supervised approaches.

References

[1] Ahmadian, A., et al. (2024). Back to Basics: Revisiting REINFORCE-Style Optimization for RLHF in LLMs. *arXiv preprint arXiv:2402.14740*.

[2] Rafailov, R., et al. (2023). Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Proceedings of NeurIPS 2023*.

[3] VanLehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist*, 46(4), 197–221.

[4] Corbett, A. T., & Anderson, J. R. (1994). Knowledge Tracing: Modelling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*, 4, 253–278.

[5] Christiano, P., et al. (2017). Deep Reinforcement Learning from Human Preferences. In *Proceedings of NeurIPS 2017*.

[6] Anderson, J. R., et al. (1995). Cognitive Tutors: Lessons Learned. *Journal of the Learning Sciences*, 4(2), 167–207.

[7] Conati, C., & VanLehn, K. (2000). Toward Computer-Based Support of Meta-Cognitive Skills: a Computational Framework to Coach Self-Explanation. *International Journal of Artificial Intelligence in Education*, 12(4), 24–43.

[8] Cobbe, K., et al. (2021). Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.

[9] Hu, E. J., et al. (2022). LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of ICLR 2022*.